**BỘ MÔN HỆ THỐNG THÔNG TIN – KHOA CÔNG NGHỆ THÔNG TIN**

**ĐẠI HỌC KHOA HỌC TỰ NHIÊN, ĐẠI HỌC QUỐC GIA TP HCM**

Nhóm sinh viên thực hiện:

1712558 Lê Phan Hoàng Linh

1712733 Trần Phạm Khánh Sơn

1712808 Trần Thị Minh Thùy

1712857 Nguyễn Anh Tú

Gíao viên phụ trách: Hồ Thị Hoàng Vy – Nguyễn Thị Như Anh

BÀI TẬP MÔN HỌC  - HỆ THỐNG THÔNG TIN PHỤC VỤ TRÍ TUỆ KINH DOANH

HỌC KỲ I – NĂM HỌC 2021-2022

**MÔN HỆ THỐNG THÔNG TIN PHỤC VỤ TRÍ TUỆ KINH DOANH**

**BẢNG PHÂN CÔNG CHI TIẾT**

|  |  |  |
| --- | --- | --- |
| **MSSV** | **Họ tên** | **Phân công công việc** |
| 1712558 | Lê Phan Hoàng Linh | * Thực hiện thiết kế DDS * Thực hiện OLAP * Thực hiện data mining |
| 1712733 | Trần Phạm Khánh Sơn | * Thực hiện thiết kế NDS và DDS * ETL dữ liệu từ Stage vào NDS * ETL dữ liệu từ NDS vào DDS * Thực hiện refresh data |
| 1712808 | Trần Thị Minh Thùy | * Thực hiện thiết kế NDS * ETL dữ liệu từ Stage vào NDS * ETL dữ liệu từ Source vào Stage * Visualize dữ liệu |
| 1712857 | Nguyễn Anh Tú | * Mô tả data, profiling data * Thiết kế + Nạp data vào NDS,DDS * Visualize , MDX * Calculated Measure |

**Mục lục**

[**I.** **NHẬN XÉT DỮ LIỆU TỪ NGUỒN:** 5](#_Toc92844717)

[1. Bảng Accidents: 5](#_Toc92844718)

[2. Bảng casualties: 7](#_Toc92844719)

[3. Bảng Vehicles 9](#_Toc92844720)

[4. Bảng Postcode 10](#_Toc92844721)

[5. Bảng LSOA 10](#_Toc92844722)

[**II.** **MÔ TẢ DỮ LIỆU:** 11](#_Toc92844723)

[1. Bảng Accident: 11](#_Toc92844724)

[2. Bảng AccidentSeverity: 12](#_Toc92844725)

[3. Bảng AgeBand: 12](#_Toc92844726)

[4. Bảng Area: 13](#_Toc92844727)

[5. Bảng Casualty: 13](#_Toc92844728)

[6. Bảng CasualtySeverity: 14](#_Toc92844729)

[7. Bảng CasualtyType: 14](#_Toc92844730)

[8. Bảng Date: 14](#_Toc92844731)

[9. Bảng JourneyPurpose: 15](#_Toc92844732)

[10. Bảng LocalAuthorityDistrict: 15](#_Toc92844733)

[11. Bảng Location: 16](#_Toc92844734)

[12. Bảng LSOA: 16](#_Toc92844735)

[13. Bảng METADATA: 17](#_Toc92844736)

[14. Bảng RoadType: 17](#_Toc92844737)

[15. Bảng GenderOfCasualty: 17](#_Toc92844738)

[16. Bảng Vehicle: 18](#_Toc92844739)

[17. Bảng VehicleType: 18](#_Toc92844740)

[**III.** **MÔ TẢ CÁC BƯỚC ĐỔ TỪ SOURCE TO STAGE:** 19](#_Toc92844741)

[1. Đối với Source CodeBook: 19](#_Toc92844742)

[2. Source Accident, Casualty, Vehicles, Postcode: 21](#_Toc92844743)

[3. Source LSOA: 24](#_Toc92844744)

[**IV.** **MÔ TẢ CÁC BƯỚC ĐỔ TỪ STAGE TO NDS:** 25](#_Toc92844745)

[1. Các bảng cần không tham chiếu khoá ngoại: 26](#_Toc92844746)

[2. Bảng LSOA: 30](#_Toc92844747)

[3. Bảng Accident: 34](#_Toc92844748)

[4. Bảng Casualty: 38](#_Toc92844749)

[5. Bảng Vehicle: 42](#_Toc92844750)

[**V.** **THIẾT KẾ NDS:** 45](#_Toc92844751)

[**VI.** **THIẾT KẾ DDS:** 46](#_Toc92844752)

[**VII.** **MÔ TẢ CÁC BƯỚC THỰC HIỆN NDS → DDS:** 47](#_Toc92844753)

[a. Các bảng Dim: 47](#_Toc92844754)

[b. Bảng FactAccident: 51](#_Toc92844755)

[c. Bảng FactCasualty: 60](#_Toc92844756)

[d. Bảng FactVehicle: 62](#_Toc92844757)

[**VIII.** **THỰC HIỆN OLAP:** 65](#_Toc92844758)

[1. Tạo phân cấp chiều thời gian: 65](#_Toc92844759)

[2. Tạo phân cấp chiều địa điểm: 67](#_Toc92844760)

[3. Tạo thêm các Named Calculation: 68](#_Toc92844761)

[3.1 AgeGroup: 68](#_Toc92844762)

[3.2 TimeOfDay: 69](#_Toc92844763)

[3.3 BuildUpRoad: 70](#_Toc92844764)

[4. Tạo Casualty cube để xem các thống kê các số liệu về thương vong: 71](#_Toc92844765)

[5. Tạo Accident cube để xem các thống kê các số liệu về TNGT: 72](#_Toc92844766)

[6. Thực hiện các thống kê: 72](#_Toc92844767)

[7. Một số query MDX cơ bản: 80](#_Toc92844768)

[**IX.** **THỰC HIỆN MINING:** 85](#_Toc92844769)

[1. Build Mining Structure: 85](#_Toc92844770)

[2. Chọn Mining Model: 86](#_Toc92844771)

[3. Xem kết quả Mining: 87](#_Toc92844772)

[3.1 Decision Tree: 87](#_Toc92844773)

[3.2 Dependency Network: 89](#_Toc92844774)

[3.3 Clustering: 90](#_Toc92844775)

[3.4 So sánh: 92](#_Toc92844776)

[**X.** **THỰC HIỆN REPORT BẰNG POWER BI:** 95](#_Toc92844777)

[1. Câu 1: 95](#_Toc92844778)

[2. Câu 2: 98](#_Toc92844779)

[3. Câu 4: 100](#_Toc92844780)

[4. Câu 5 100](#_Toc92844781)

[5. Câu 6: 102](#_Toc92844782)

[6. Câu 7: 103](#_Toc92844783)

[7. Câu 8: 104](#_Toc92844784)

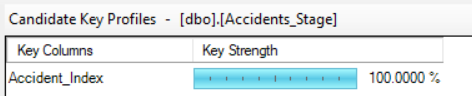
[8. Câu 9: 104](#_Toc92844785)

[9. Câu 11: 106](#_Toc92844786)

[**XI.** **LẬP LỊCH THAO TÁC TỰ ĐỘNG ETL** 107](#_Toc92844787)

# **NHẬN XÉT DỮ LIỆU TỪ NGUỒN:**

## Bảng Accidents:



Độ mạnh khóa của các thuộc tính trong bảng cho thấy, cột Accident\_Index có độ mạnh khóa là 100% vì vậy ta có thể xác định đây là khóa chính tự nhiên của bảng.

A picture containing chart

Description automatically generated

Kiểm tra tính NULL trên bảng, ta có thể thấy với bảng Accident hầu hết các cột đều có giá trị chỉ có hai cột là LSOA\_of\_Accidents\_Location và time có giá trị NULL với số lượng cụ thể từng cột là 38691 và 25.

Graphical user interface

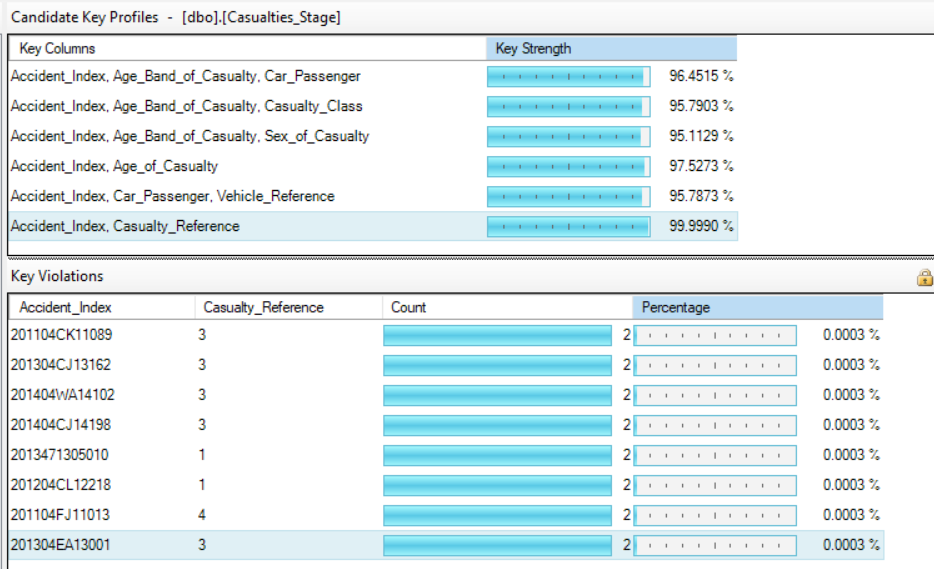
Description automatically generated

Kiểm tra tính duy nhất của các giá trị trên bảng chúng ta có thể thấy tại cột LSOA\_of\_Accident\_Location có 34865 giá trị duy nhất có thể suy ra có 34865 địa điểm xảy ra tai nạn.Graphical user interface, text, application

Description automatically generated

Kiểm tra min max trên các giá trị của bảng chúng ta có thể thấy tại cột date cho ta khoảng thời gian ghi nhận dữ liệu của nguồn từ 1/1/2011 tới 31/12/2014.

## Bảng casualties:



Mỗi dòng Accident\_Index không thể xác định duy nhất 1 dòng dữ liệu trong Casualty, nên ta chọn cặp (Accident\_Index, Casualty\_Reference) làm khóa chính. Đồng thời xóa bỏ các trường hợp có Accident\_Index bị duplicate khóa chính (Có thể xuất phát từ lỗi đánh máy)

Chart, funnel chart

Description automatically generated

Kiểm tra mức độ lặp lại của các giá trị của bảng với cột Age\_Band\_Casualty ta có thể thấy nhóm tuổi tai nạn chủ yếu là những người trưởng thành 6-8(26-55 tuổi), và thanh thiếu niên 4-5(16-25 tuổi)

Graphical user interface, application

Description automatically generated

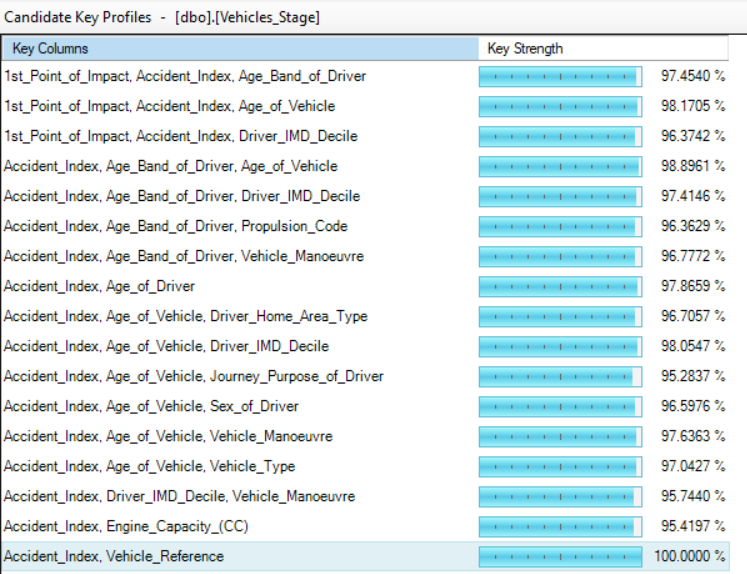
Kiểm tra mức độ lặp lại của các giá trị của bảng với cột Casualty\_Type, ta có thể thấy các tai nạn chủ yếu là xe hơi, chiếm hơn 58% (ứng với Type=9)

Graphical user interface, application

Description automatically generated

Kiểm tra mức độ lặp lại của các giá trị của bảng với cột Casualty\_Severity ta có thể thấy ở mức Casualty\_Severity = 3 là cao nhất ứng với mức độ tại nạn nhẹ chiếm 87.43%.

## Bảng Vehicles



Độ mạnh khóa của các thuộc tính trong bảng cho thấy, cặp (Accident\_Index, Vehicle\_Referenc) có độ mạnh khóa là 100% vì vậy ta có thể xác định đây là khóa chính tự nhiên của bảng.

Table

Description automatically generated

Kiểm tra tính NULL của các thuộc tính trong bảng ta thấy bảng không có dòng dữ liệu nào nhận giá trị NULL ở tất cả các cột.

## Bảng Postcode

Table

Description automatically generated

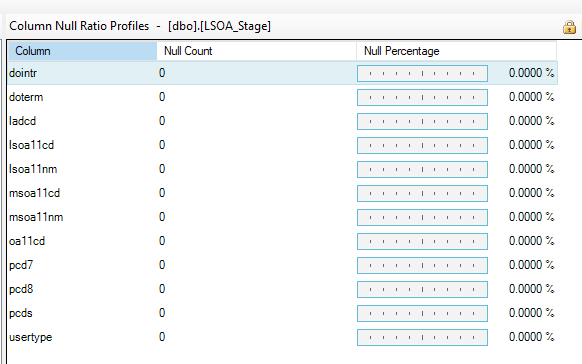
 Độ mạnh khóa của các thuộc tính trong bảng cho thấy, cột postcode có độ mạnh khóa là 100% vì vậy ta có thể xác định đây là khóa chính của bảng

## Bảng LSOA

A picture containing chart

Description automatically generated

Độ mạnh khóa của các thuộc tính trong bảng cho thấy, cột pcd7, pcd8, pcds có độ mạnh khóa là 100% vì vậy ta có thể xác định đây có thể là các cột có thể được dùng làm khóa chính của bảng.



Kiểm tra tính NULL của các thuộc tính trong bảng ta thấy bảng không có dòng dữ liệu nào nhận giá trị NULL ở tất cả các cột.

# **MÔ TẢ DỮ LIỆU:**

## Bảng Accident:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | AccidentID | int | Khóa tự tăng |
| 2 | AccidentIndex | varchar(50) | Mã tai nạn |
| 3 | AccidentSeverityID | int | Mức độ nghiêm trọng của tai nạn  (1: Fatal, 2: Serious, 3: Slight) |
| 4 | NumberOfVehicles | int | Số lượng phương tiện |
| 5 | NumberOfCasualties | int | Số lượng người chết |
| 6 | LocalAuthorityDistrictID | int | Địa phương |
| 7 | Time | time(7) | Thời gian xảy ra tai nạn |
| 8 | Date | date | Ngày xảy ra tai nạn |
| 9 | RoadTypeID | int | Loại đường |
| 10 | SpeedLimit | int | Tốc độ tối đa |
| 11 | AreaID | int | Thành thị/Nông thôn |
| 12 | LocationID | int | Địa điểm xảy ra tai nạn |
| 13 | SourceID | int | Khóa nguồn dữ liệu |
| 14 | CreatedDate | Datetime | Ngày thêm |
| 15 | UpdatedDate | Datetime | Ngày cập nhật |

## Bảng AccidentSeverity:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | AccidentSeverityID | int | Khóa tự tăng |
| 2 | AccidentSeverityCode | int | Mã mức độ nghiêm trọng của tai nạn |
| 3 | AccidentSeverityLabel | nvarchar(255) | Nhãn mức độ nghiêm trọng của tai nạn |
| 4 | SourceID | int | Khóa nguồn dữ liệu |
| 5 | CreatedDate | Datetime | Ngày thêm |
| 6 | UpdatedDate | Datetime | Ngày cập nhật |

## Bảng AgeBand:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | AgeBandID | int | Khóa tự tăng |
| 2 | AgeBandode | int | Mã nhóm tuổi |
| 3 | AgeBandLabel | nvarchar(255) | Nhãn nhóm tuổi |
| 4 | SourceID | int | Khóa nguồn dữ liệu |
| 5 | CreatedDate | Datetime | Ngày thêm |
| 6 | UpdatedDate | Datetime | Ngày cập nhật |

## Bảng Area:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | AreaID | int | Khóa tự tăng |
| 2 | AreaCode | int | Mã khu vực nông thôn hay thành thị |
| 3 | AreaLabel | nvarchar(255) | Nhãn khu vực nông thôn hay thành thị |
| 4 | SourceID | int | Khóa nguồn dữ liệu |
| 5 | CreatedDate | Datetime | Ngày thêm |
| 6 | UpdatedDate | Datetime | Ngày cập nhật |

## Bảng Casualty:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | CasualtyID | int | identity |
| 2 | AccidentID | varchar(50) | Mã tai nạn |
| 3 | CasualtyReference | int | mã nạn nhân tham chiếu |
| 4 | GenderID | int | giới tính nạn nhân |
| 5 | Age | int | độ tuổi của nạn nhân |
| 6 | AgeBandID | int | nhóm tuổi của nạn nhân |
| 7 | CasualtySeverityID | int | mức độ nghiêm trọng |
| 8 | CasualtyTypeID | int | loại nạn nhân |
| 9 | SourceID | int | Khóa nguồn dữ liệu |
| 10 | CreatedDate | datetime | Thời gian tạo bảng |
| 11 | UpdatedDate | datetime | Thời gian cập nhật bảng |

## Bảng CasualtySeverity:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | CasualtySeverityID | int | Khóa tự tăng |
| 2 | CasualtySeverityCode | int | Mã thương vong |
| 3 | CasualtySeverityLabel | nvarchar(255) | Nhãn thương vong |
| 4 | SourceID | int | Khóa nguồn dữ liệu |
| 5 | CreatedDate | Datetime | Ngày thêm |
| 6 | UpdatedDate | Datetime | Ngày cập nhật |

## Bảng CasualtyType:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | CasualtyTypeID | int | Khóa tự tăng |
| 2 | CasualtyTypeCode | int | Mã loại thương vong |
| 3 | CasualtyTypeLabel | nvarchar(255) | Nhãn loại thương vong |
| 4 | SourceID | int | Khóa nguồn dữ liệu |
| 5 | CreatedDate | Datetime | Ngày thêm |
| 6 | UpdatedDate | Datetime | Ngày cập nhật |

## Bảng Date:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | DateID | int | Khóa tự tăng |
| 2 | Date | date | Ngày đầy đủ |
| 3 | Day | int | Giá trị ngày |
| 4 | Month | int | Giá trị tháng |
| 5 | Quarter | int | Giá trị quý |
| 6 | Year | int | Giá trị năm |
| 7 | SourceID | int | Khóa nguồn dữ liệu |
| 8 | CreatedDate | Datetime | Ngày thêm |
| 9 | UpdatedDate | Datetime | Ngày cập nhật |

## Bảng JourneyPurpose:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | JourneyPurposeID | int | Khóa tự tăng |
| 2 | JourneyPurposeCode | int | Mã mục đích hành trình |
| 3 | JourneyPurposeLabel | nvarchar(255) | Nhãn mục đích hành trình |
| 4 | SourceID | int | Khóa nguồn dữ liệu |
| 5 | CreatedDate | Datetime | Ngày thêm |
| 6 | UpdatedDate | Datetime | Ngày cập nhật |

## Bảng LocalAuthorityDistrict:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | LocalAuthorityDistrictID | int | Khóa tự tăng |
| 2 | LocalAuthorityDistrictCode | int | Mã vị trí |
| 3 | LocalAuthorityDistrictLabel | nvarchar(255) | Nhãn vị trí |
| 4 | SourceID | int | Khóa nguồn dữ liệu |
| 5 | CreatedDate | Datetime | Ngày thêm |
| 6 | UpdatedDate | Datetime | Ngày cập nhật |

## Bảng Location:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | LocationID | int | Khóa tự tăng |
| 2 | postcode | varchar(50) | Mã bưu điện |
| 3 | city | varchar(50) | Tên thành phố |
| 4 | county | varchar(50) | Tên hạt |
| 5 | country | varchar(50) | Tên quốc gia |
| 6 | region | varchar(50) | Tên vùng |
| 7 | SourceID | int | Khóa nguồn dữ liệu |
| 8 | CreatedDate | Datetime | Ngày thêm |
| 9 | UpdatedDate | Datetime | Ngày cập nhật |

## Bảng LSOA:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | LSOAID | int | Khóa tự tăng |
| 2 | LocationID | int | Mã vị trí |
| 3 | lsoa11cd | varchar(50) |  |
| 4 | SourceID | int | Khóa nguồn dữ liệu |
| 5 | CreatedDate | Datetime | Ngày thêm |
| 6 | UpdatedDate | Datetime | Ngày cập nhật |

## Bảng METADATA:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | [index] | int | Khóa tự tăng |
| 2 | CreatedDate | Datetime | Ngày thêm |
| 3 | UpdatedDate | Datetime | Ngày cập nhật |

## Bảng RoadType:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | RoadTypeID | int | Khóa tự tăng |
| 2 | RoadTypeCode | int | Mã loại đường |
| 3 | RoadTypeLabel | nvarchar(255) | Nhãn loại đường |
| 4 | SourceID | int | Khóa nguồn dữ liệu |
| 5 | CreatedDate | Datetime | Ngày thêm |
| 6 | UpdatedDate | Datetime | Ngày cập nhật |

## Bảng GenderOfCasualty:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | GenderID | int | Khóa tự tăng |
| 2 | GenderCode | int | Mã giới tính |
| 3 | GenderLabel | nvarchar(255) | Nhãn giới tính |
| 4 | SourceID | int | Khóa nguồn dữ liệu |
| 5 | CreatedDate | Datetime | Ngày thêm |
| 6 | UpdatedDate | Datetime | Ngày cập nhật |

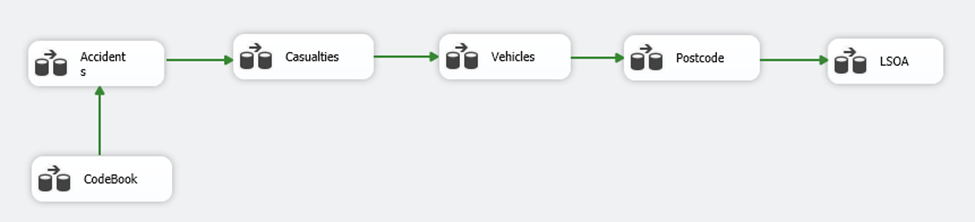
## Bảng Vehicle:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | VehicleID | int | Identity |
| 2 | AccidentID | int | Mã tai nạn |
| 3 | VehicleReference | int | Mã phương tiện tham chiếu: phương tiện đầu tiên được mã hóa phải được đánh số 001 và bất kỳ  phương tiện nào tiếp theo phải được đánh số theo  thứ tự |
| 4 | VehicleTypeID | int | Loại phương tiện |
| 5 | JourneyPurposeID | int | Mục đích hành trình của người lái xe |
| 6 | SourceID | int | Khóa nguồn dữ liệu |
| 7 | CreatedDate | datetime | Thời gian tạo bảng |
|  | UpdatedDate | datetime | Thời gian cập nhật bảng |

## Bảng VehicleType:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thuộc tính** | **Kiểu dữ liệu** | **Ý nghĩa** |
| 1 | VehicleTypeID | int | Khóa tự tăng |
| 2 | VehicleTypeCode | int | Mã loại phương tiện |
| 3 | VehicleTypeLabel | nvarchar(255) | Nhãn loại phương tiện |
| 4 | SourceID | int | Khóa nguồn dữ liệu |
| 5 | CreatedDate | Datetime | Ngày thêm |
| 6 | UpdatedDate | Datetime | Ngày cập nhật |

# **MÔ TẢ CÁC BƯỚC ĐỔ TỪ SOURCE TO STAGE:**



Ở bước này, vì chưa phải chịu ràng buộc giữa các bảng nên có thể đổ thứ tự các bảng tùy ý.

## 1. Đối với Source CodeBook:

Graphical user interface

Description automatically generated

Các bảng trong codebook sẽ tương tự nhau, lấy ví dụ bảng AccidentSeverity:

Diagram

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **STT** | **Các bước** | **Ý nghĩa** |
| 1 | Kéo 1 Excel Source | Lấy dữ liệu từ nguồn excel  Graphical user interface, text, application  Description automatically generated  tại đây ta sẽ chọn nguồn như hình bên bằng cách chọn file nguồn,  Chart  Description automatically generated with low confidence  Phần tiếp sẽ chọn bảng mình muốn chọn rồi chọn các cột muốn lấy tại option column Graphical user interface  Description automatically generated with medium confidence |
| 2 | Kéo 1 derived Column | Thực hiện thay thế cột code cũ có kiểu dữ liệu float thành cột code mới kiểu integer => mục đích dễ dàng sau này tạo liên kết giữa các bảng qua cột code |
| 3 | Kéo 1 OLE DB destination | Dùng để chọn đích,  Graphical user interface, text, application, website  Description automatically generated  Chọn database và bảng muốn chọn làm bảng đích  Diagram  Description automatically generated  Mapping lại các cột cho phù hợp |

## 2. Source Accident, Casualty, Vehicles, Postcode:



Lấy ví dụ cho bảng Accidents Source, sẽ chỉ có 2 bước do 2 bảng này không cần thay đổi dữ liệu

|  |  |  |
| --- | --- | --- |
| **STT** | **Các bước** | **Ý nghĩa** |
| 1 | Kéo Flat File Source | Lấy dữ liệu từ nguồn  Graphical user interface, text, application  Description automatically generated  Tại đây ta sẽ chọn nguồn như hình bên bằng cách chọn file nguồn,  Phần tiếp sẽ chọn bảng mình muốn chọn rồi chọn các cột muốn lấy tại option column  Graphical user interface, application, Word  Description automatically generated  Table  Description automatically generated |
| 2 | Kéo 1 OLE DB destination | Dùng để chọn đích,  Graphical user interface, text, application, email  Description automatically generated  Chọn database và bảng muốn chọn làm bảng đích  Mapping lại các cột cho phù hợp  Table  Description automatically generated |

## 3. Source LSOA:

Diagram

Description automatically generated

Cũng giống các bảng ở bước trên, chỉ có thêm 1 bước ở giữa là :

Kéo Script Component :

- Tại Script language chọn ngôn ngữ viết, ở đây nhóm em chọn C# rồi chọn edit script :



Graphical user interface, application

Description automatically generated

- Chọn input column để chọn các cột cần lấy để biến đổi

Graphical user interface, diagram

Description automatically generated with medium confidence

- Chọn add ouput column để tạo cột ouput và kiểu dữ liệu :

Graphical user interface, application

Description automatically generated

- Viết hàm phù hợp để cắt chuỗi:

Split\_pcds: bỏ dấu “ đầu tiên, láy các kí tự trc dấu cách

splitlsoa11cd: Bỏ dấu " đầu tiên, lấy các kí tự trước dấu " tiếp theo

# **MÔ TẢ CÁC BƯỚC ĐỔ TỪ STAGE TO NDS:**

Graphical user interface, application

Description automatically generated

Để thực hiện được việc đổ dữ liệu từ Stage vào NDS ta cần chú ý đến khoá ngoại của các bảng trong NDS, điều này sẽ quyết định thứ tự đổ dữ liệu của các bảng trong NDS cụ thể như sau:

## Các bảng cần không tham chiếu khoá ngoại:

|  |  |  |
| --- | --- | --- |
| **STT** | **Các bước** | **Ý nghĩa** |
| 1 | Data flow task  (Với bảng Area) | Đổ dữ liệu từ Stage -> NDS |
| 3 | Nạp từ stage vào bảng Area trong NDS |  |
| 4 | Chọn nguồn dữ liệu | Ta chọn nguồn dữ liệu nạp vào là bảng Area trong Stage:  Lưu ý chọn đúng kết nối tại mục OLE DB Connection manager cũng như đúng tên bảng tại mục Name of the table |
| 5 | Thêm cột nguồn  dùng derived column | Ta thêm tên cột muốn chèn, chọn kiểu chèn là thêm cột mới <add as new column> và giá trị muốn chèn vào tại cột Expression |
| 6 | Thực hiện kiểm tra lookup giá trị trong stage và NDS | Tại phần Connection chọn ta lưu ý chọn đúng kết nối tại mục OLE DB Connection manager cũng như đúng tên bảng tại mục Name of the table, ở đây là ở trong NDS chứ không phải trong stage    Tại phần Column, ta chọn giá trị sẽ được lookup cũng như giá trị được thêm vào, ở đây cụ thể ta sẽ thực hiện so sánh cột code ở trong Area\_stage với AreaCode trong Area\_NDS và thêm cột AreaID vào để có thể tiến hành thực hiện việc lookup có điều kiện. |
| 7 | Lookup match giá trị phù hợp | Khi match giá trị phù hợp ta thực hiện thêm cột mới UpdateDate, ở đây có nghĩa là giá trị này đã là giá trị cũ chỉ thực hiện update    Sau đó ta thực hiện cập nhật bảng area trong NDS với lệnh SQL sau    Với các giá trị tham số mapping các cột: |
| 8 | Lookup unmatch giá trị phù hợp | Khi unmatch giá trị phù hợp ta thực hiện thêm cột mới UpdateDate, và cả CreatedDate, ở đây có nghĩa là giá trị này là giá trị mới nên ta thực hiện thêm mới.      Ta thực hiện thêm mới các giá trị vào bảng Area trong NDS với mapping cụ thể như sau: |

Đối với các bảng không tham chiếu khoá ngoại khác cụ thể như hình dưới ta thực hiện việc đổ dữ liệu từ stage vào NDS một cách tương tự

Graphical user interface, diagram

Description automatically generated

## Bảng LSOA:

|  |  |  |
| --- | --- | --- |
| **STT** | **Các bước** | **Ý nghĩa** |
| 1 | Data flow task  (Với bảng LSOA) | Đổ dữ liệu từ Stage -> NDS  Bảng này được đổ sau khi đổ bảng Location vì chúng có liên hệ khoá ngoại |
| 2 | Nạp từ stage vào bảng LSOA trong NDS | Ta thực hiện tương tự như đổ các bảng không có tham chiếu khoá ngoại tuy nhiên ta cần phải thực hiện Lookup lần lượt các khoá ngoại tồn tại. Ở trong bảng LSOA cụ thể là Geography    Tại mục Connection chọn kết nối cũng như bảng cần kiểm tra    Tại mục Column, ta thực hiện match pcdsspilit với postcode để thực hiện việc so sánh, thực hiện thêm cột LocationID vào bảng. |
| 3 | Thực hiện kiểm tra lookup giá trị trong stage và NDS | Tại phần Connection chọn ta lưu ý chọn đúng kết nối tại mục OLE DB Connection manager cũng như đúng tên bảng tại mục Name of the table, ở đây là ở trong NDS chứ không phải trong stage    Tại phần Column, ta chọn giá trị sẽ được lookup cũng như giá trị được thêm vào, ở đây cụ thể ta sẽ thực hiện so sánh cột Location ở trong Location\_stage với LocationID trong LSOA\_NDS và thêm cột LSOAID và LocationID vào để có thể tiến hành thực hiện việc lookup có điều kiện. |
| 4 | Lookup unmatch giá trị phù hợp | Khi unmatch giá trị phù hợp ta thực hiện thêm cột mới UpdateDate, và CreatedDat ở đây có nghĩa là giá trị này là giá trị mới nên thực hiện việc thêm mới vào LSOA\_NDS  Ta thực hiện thêm mới vào bảng LSOA\_NDS với các giá trị tham số mapping các cột: |

## Bảng Accident:

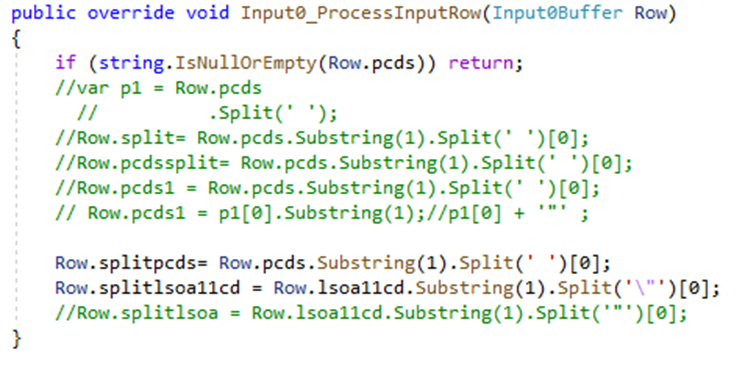
|  |  |  |
| --- | --- | --- |
| **STT** | **Các bước** | **Ý nghĩa** |
| 1 | Data flow task  (Với bảng Accident) | Đổ dữ liệu từ Stage -> NDS |
| 2 | Nạp từ stage vào bảng Accident trong NDS | Ta thực hiện tương tự như đổ các bảng không có tham chiếu khoá ngoại tuy nhiên ta cần phải thực hiện Lookup lần lượt các khoá ngoại tồn tại. Ở trong bảng Accident cụ thể là LSOA\_NDS, AccidentSeverity\_NDS, LocalAuthorityDistrict\_NDS, RoadType\_NDS, Area\_NDS  Tại mục Connection chọn kết nối cũng như tên bảng cần kiểm tra  Tại mục Column, ta thực hiện match các cột tương ứng để thực hiện việc so sánh, thực hiện thêm cột các cột ID tương ứng vào bảng cụ thể theo các hình dưới:  Với LSOA\_NDS    Với AccidentSeverity\_NDS    Với LocalAuthorityDistrict\_NDS    Với RoadType\_NDS    Với Area\_NDS |
| 4 | Thực hiện kiểm tra lookup giá trị trong stage và NDS và thêm mới hoặc cập nhật trong NDS | Ta thực hiện tương tự các bảng trên theo luồng bên dưới    Đặc biệt có phần Check real change sẽ thực hiện việc lookup tất cả các các thuộc tính của bảng để kiểm tra xem có thực sự thay đổi so với dữ liệu đã có, nếu không thoả mới thực hiện việc update với tham số cụ thể của các column như sau |

## Bảng Casualty:

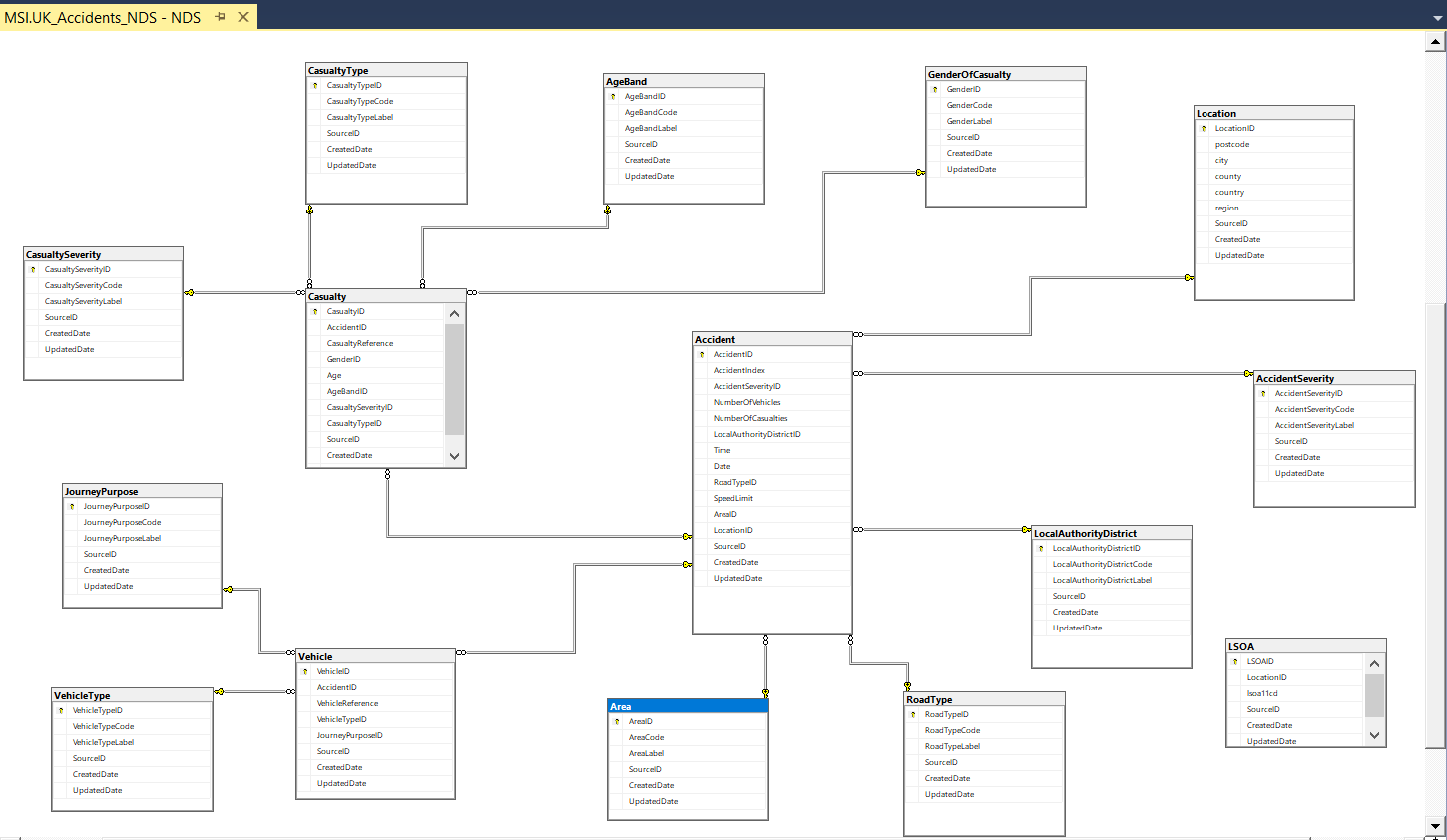
|  |  |  |
| --- | --- | --- |
| **STT** | **Các bước** | **Ý nghĩa** |
| 1 | Data flow task  (Với bảng Casualty) | Đổ dữ liệu từ Stage -> NDS |
| 2 | Nạp từ stage vào bảng Casualty trong NDS | Ta thực hiện tương tự như đổ các bảng không có tham chiếu khoá ngoại tuy nhiên ta cần phải thực hiện Lookup lần lượt các khoá ngoại tồn tại. Ở trong bảng Casualty cụ thể là: Accident\_NDS, Gender\_NDS, AgeBand\_NDS, CasualtySeverity\_NDS,CasualtyType\_NDS  Tại mục Connection chọn kết nối cũng như tên bảng cần kiểm tra  Tại mục Column, ta thực hiện match các cột tương ứng để thực hiện việc so sánh, thực hiện thêm cột các cột ID tương ứng vào bảng cụ thể theo các hình dưới:  Với Accident\_NDS:    Gender\_NDS:    Với AgeBand\_NDS:    Với CasualtySeverity\_NDS    Với CasualtyType\_NDS |
| 4 | Thực hiện kiểm tra lookup giá trị trong stage và NDS và thêm mới hoặc cập nhật trong NDS | Ta thực hiện tương tự các bảng trên theo luồng bên dưới    Đặc biệt có phần Check real change sẽ thực hiện việc lookup tất cả các các thuộc tính của bảng để kiểm tra xem có thực sự thay đổi so với dữ liệu đã có, nếu không thoả mới thực hiện việc update với tham số cụ thể của các column như sau |

## Bảng Vehicle:

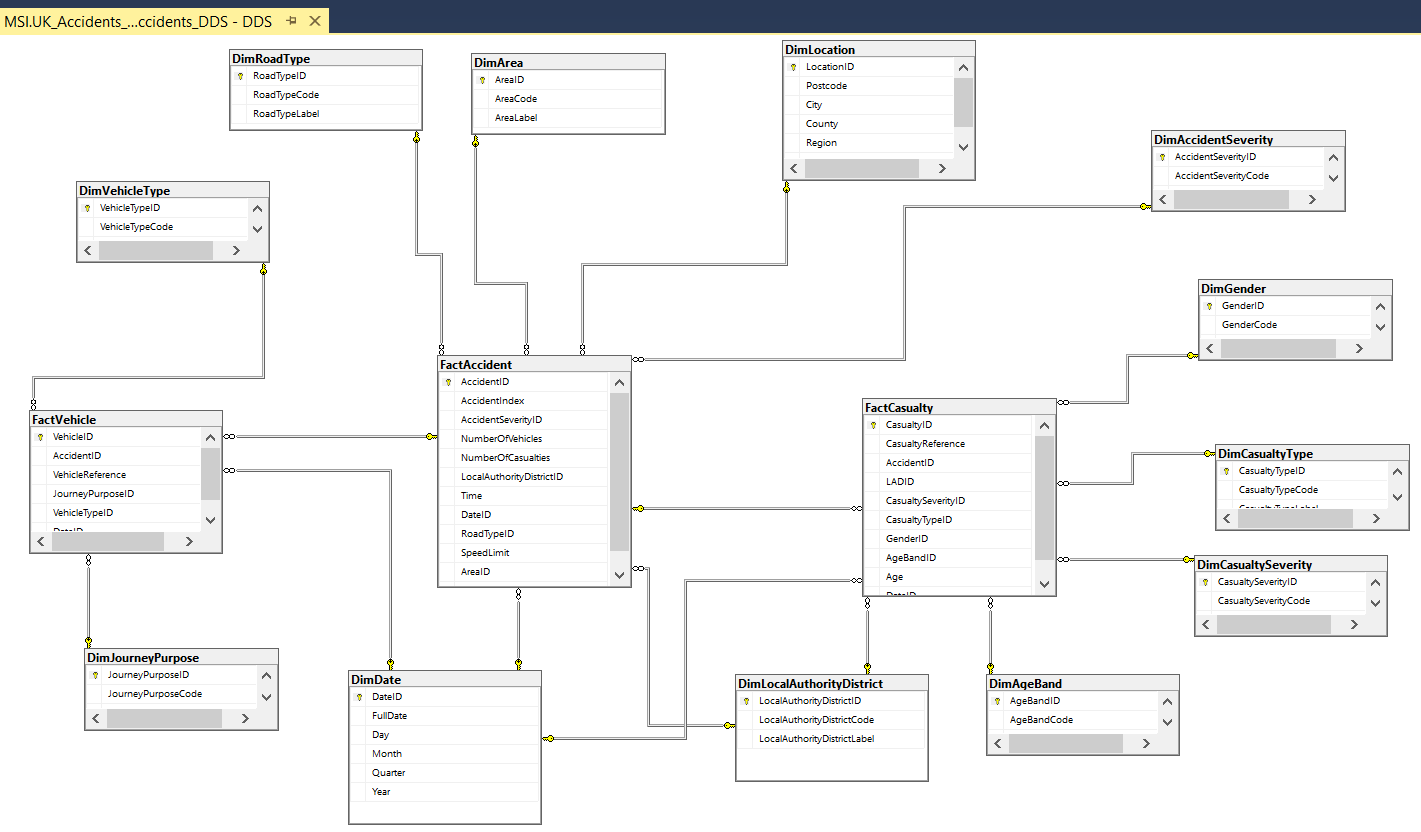
|  |  |  |
| --- | --- | --- |
| **STT** | **Các bước** | **Ý nghĩa** |
| 1 | Data flow task  (Với bảng Vehicle) | Đổ dữ liệu từ Stage -> NDS |
| 2 | Nạp từ stage vào bảng Casualty trong NDS | Ta thực hiện tương tự như đổ các bảng không có tham chiếu khoá ngoại tuy nhiên ta cần phải thực hiện Lookup lần lượt các khoá ngoại tồn tại. Ở trong bảng Casualty cụ thể là: Accident\_NDS, VehicleType\_NDS, JourneyPurpose\_NDS  Tại mục Connection chọn kết nối cũng như tên bảng cần kiểm tra  Tại mục Column, ta thực hiện match các cột tương ứng để thực hiện việc so sánh, thực hiện thêm cột các cột ID tương ứng vào bảng cụ thể theo các hình dưới:  Với Accident\_NDS:    Với VehicleType\_NDS:    Với JourneyPurpose\_NDS: |
| 4 | Thực hiện kiểm tra lookup giá trị trong stage và NDS và thêm mới hoặc cập nhật trong NDS | Ta thực hiện tương tự các bảng trên theo luồng bên dưới    Đặc biệt có phần Check real change sẽ thực hiện việc lookup tất cả các các thuộc tính của bảng để kiểm tra xem có thực sự thay đổi so với dữ liệu đã có, nếu không thoả mới thực hiện việc update với tham số cụ thể của các column như sau |



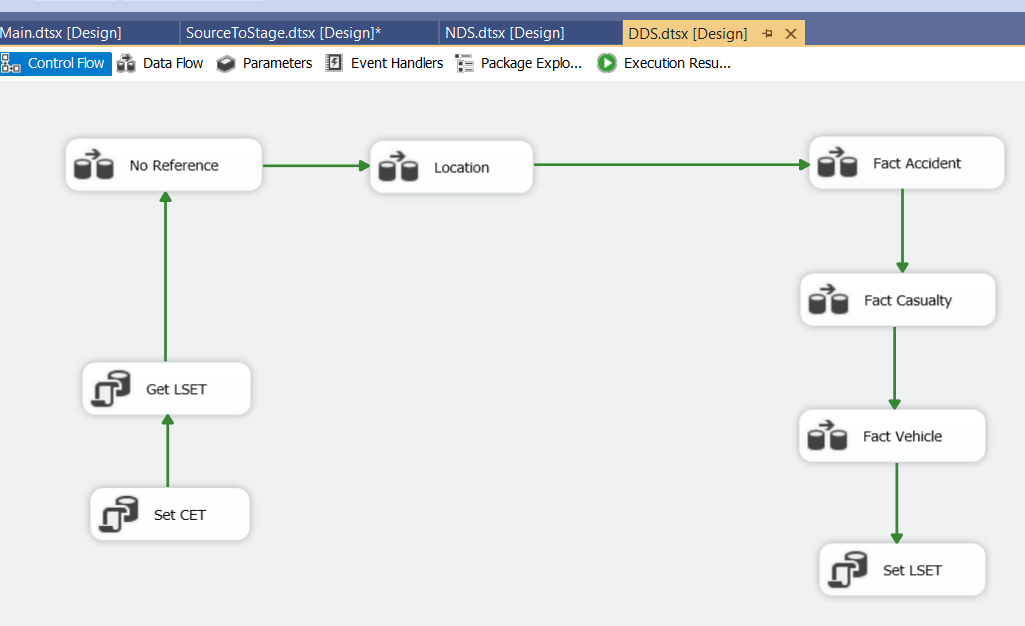
# **THIẾT KẾ NDS:**



# **THIẾT KẾ DDS:**



# **MÔ TẢ CÁC BƯỚC THỰC HIỆN NDS → DDS:**



Lưu lại thời gian thực hiện nạp data từ NDS → DDS

## Các bảng Dim:

Không cần kiểm tra khóa ngoại, cài đặt SCD (Slowly changing dimension), thực hiện ghi đè lên những giá trị cũ. \_Code sẽ là BusinessKey, \_Label sẽ là giá trị được ghi đè.

|  |  |  |
| --- | --- | --- |
| **STT** | **Các bước** | **Ý nghĩa** |
| 1 | Set CET  Get LSET | Set giá trị CET    Lấy giá trị LSET, CET của bảng Data\_flow trong Metadata (chứa thông tin thời gian ETL)    Gán 2 biến CET, LSET ở trên thành 2 biến User::CET, User::LSET |
| 2 | Data flow task  (Reference) | Đổ dữ liệu từ NDS → DDS |
| 3 | Nạp vào bảng chiều Local Authority District trong NDS | Lấy thông tin từ bảng NDS với các điều kiện từ biến LSET, CET đã được lấy ra |
| 4 | Slowly Changing Dimension: Thực hiện ghi đè lên giá trị cũ trong bảng của DDS nếu có giá trị cập nhật | Chọn connection, table, Business key sau đó nextChọn thuộc tính trong bảng chiều, loại thay đổi (SQL server hỗ trợ 3 kiểu: Fixed Attribute (Không cho thay đổi giá trị), Changing Attribute (Ghi đè lên giá trị thay đổi), Historial Attribute (Lưu lịch sử giá trị thay đổi)  Sau đó bấm Next để được generate các luồng đi mặc định (có thể chỉnh phù hợp với nhu cầu) |
| 5 | Thực hiện tương tự với các bảng Dim khác |  |

## Bảng FactAccident:

|  |  |  |
| --- | --- | --- |
| **STT** | **Các bước** | **Ý nghĩa** |
| 1 | Luồng xử lý |  |
| 2 | Nạp vào bảng FactAccident trong DDS | Lấy thông tin từ bảng Accident trong NDS với các điều kiện từ biến LSET, CET đã được lấy ra |
| 3 | Kiểm tra khóa ngoại của các bảng Dim đã có trong DDS | Chọn Lookup, sau đỏ chọn Connection, chọn DimTable, chọn mục General, chọn Ignore failure như trên hình    Sau đó chọn mục Columns, Mapping 2 thuộc tính (1 thuộc tính trong Accident NDS, 1 thuộc tính trong bảng Dim) |
| 4 | Thực hiện kiểm tra khóa ngoại tương tự với các bảng Dim khác | DimLocalAuthorityDistrict, DimRoadType, DimUrbanArea, DimLocation |
| 5 | Thêm cột bằng Derived Column |  |
| 6 | Kiểm tra khóa ngoại đối với bảng DimDate | Cũng tương tự như ở trên, tuy nhiên lúc mapping thì điều chỉnh các tham số như trên hình |
| 7 | Kiểm tra dòng dữ liệu này đã tồn tại trong FactAccident hay chưa | Chọn connection, bảng FactAccident    Mapping 2 cột AccidentID (trong NDS với DDS) để kiểm tra đã tồn tại hay chưa, sau đó tích chọn AccidentID (để dùng lại ở bước sau) |
| 8 | Chia làm 2 luồng: Nếu AccidentID đó chưa có trong DDS thì tiến hành Insert (đường màu đỏ)  Ngược lại, thì cho dữ liệu qua luồng màu xanh |  |
| 8.1 | Luồng màu đỏ  (Lookup Error Output) | Chọn connection, bảng FactAccident    Sau đó kiểm tra bảng mapping xem đã đúng hay chưa, bấm OK để hoàn tất |
| 8.2 | Luồng màu xanh  (Look Match Output) | Vì dữ liệu đã có tồn tại, nên chúng ta tiếp tục kiểm tra dữ liệu có thực sự thay đổi hay không, nếu không thì bỏ qua, nếu có thì mới update      Kiểm tra những thuộc tính có giống trong DDS hay không:  Nếu có thì bỏ qua, không thực hiện nữa  Nếu không thì tiến hành update (Luồng Lookup No Match Output)    Chọn connection    Chọn Component Properties, ấn vào ‘...’, sau đó viết script update trong FactAccident, đối với dấu ? thì qua mục Column Mappings để chỉnh    Chọn Column Mappings, sau đó mapping theo thứ tự dấu ? ở bên Component, ấn OK để hoàn tất |

## Bảng FactCasualty:

|  |  |  |
| --- | --- | --- |
| **STT** | **Các bước** | **Ý nghĩa** |
| 1 | Luồng xử lý |  |
| 2 | Kiểm tra khóa ngoại | Ta cũng tiến hành nạp dữ liệu từ bảng Casualty trong NDS vào với điều kiện như ở trên, Sau đó tiến hành kiểm tra khóa ngoại như FactAccident. Một điều khác biệt ở đây là kiểm tra thêm AccidentID trong bảng FactAccident    Chọn Connection, lấy ra những thuộc tính như trong hình    So sánh khóa ngoại AccidentID của bảng Casualty trong NDS với FactAccident trong DDS, sau đó lấy thêm 2 thuộc tính là LocalAuthorityDistrictID và DateID để phục vụ nhu cầu truy vấn của đề bài đưa ra |

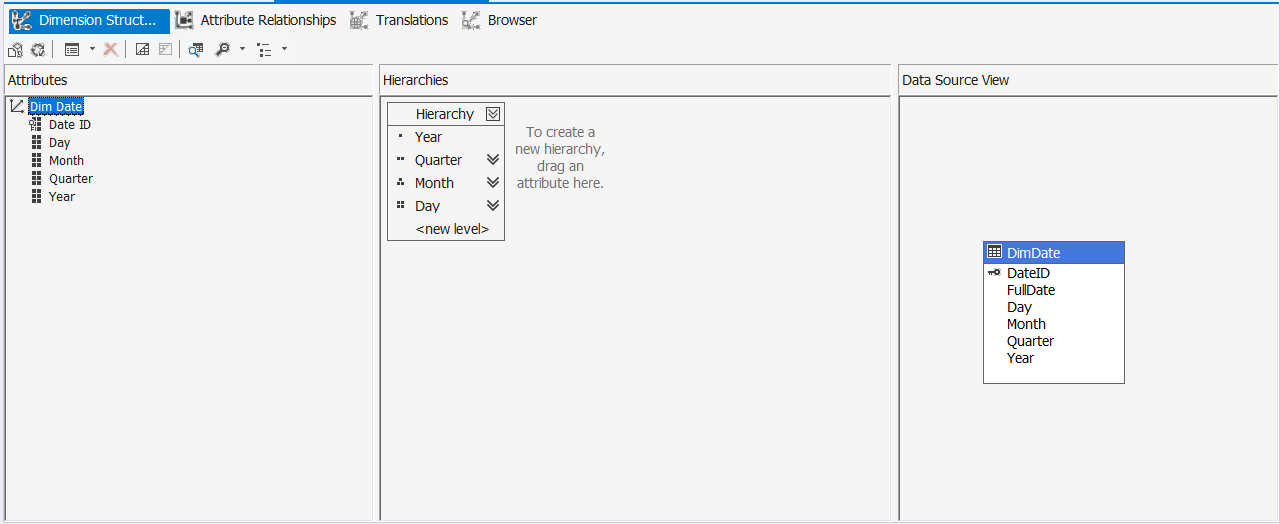
## Bảng FactVehicle:

|  |  |  |
| --- | --- | --- |
| **STT** | **Các bước** | **Ý nghĩa** |
| 1 | Luồng xử lý |  |
| 2 | Kiểm tra khóa ngoại | Ta cũng tiến hành nạp dữ liệu từ bảng Vehicle trong NDS vào với điều kiện như ở trên, Sau đó tiến hành kiểm tra khóa ngoại như FactAccident. Một điều khác biệt ở đây là kiểm tra thêm AccidentID trong bảng FactAccident    Chọn Connection, lấy ra những thuộc tính như trong hình    So sánh khóa ngoại AccidentID của bảngVehicle trong NDS với FactAccident trong DDS, sau đó lấy thêm 2 thuộc tính là DateID để phục vụ nhu cầu truy vấn của đề bài đưa ra |

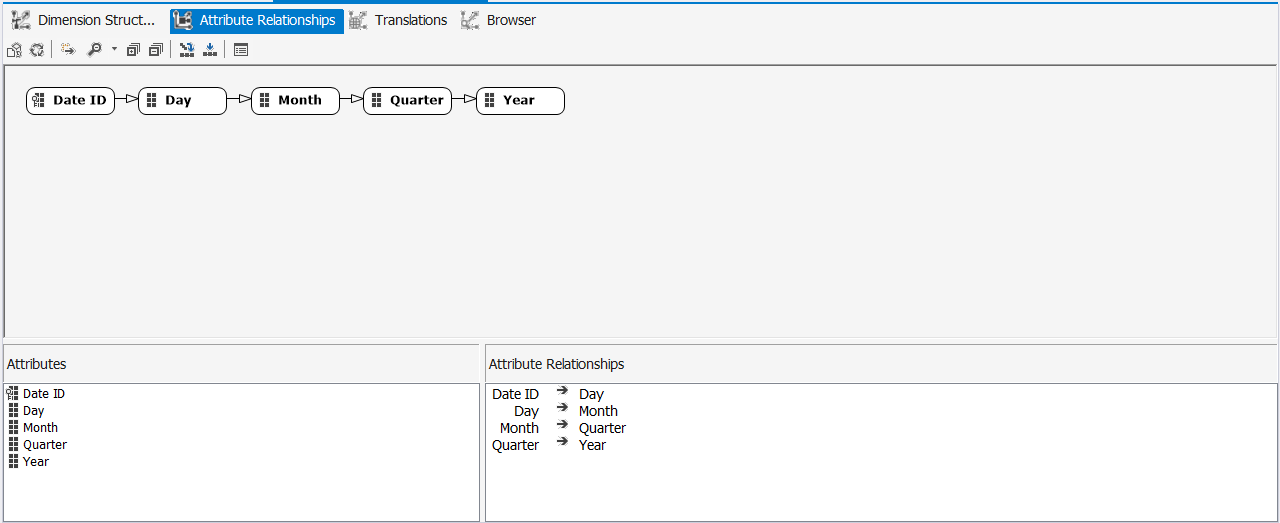
# **THỰC HIỆN OLAP:**

## Tạo phân cấp chiều thời gian:

* Tạo bảng kế thừa

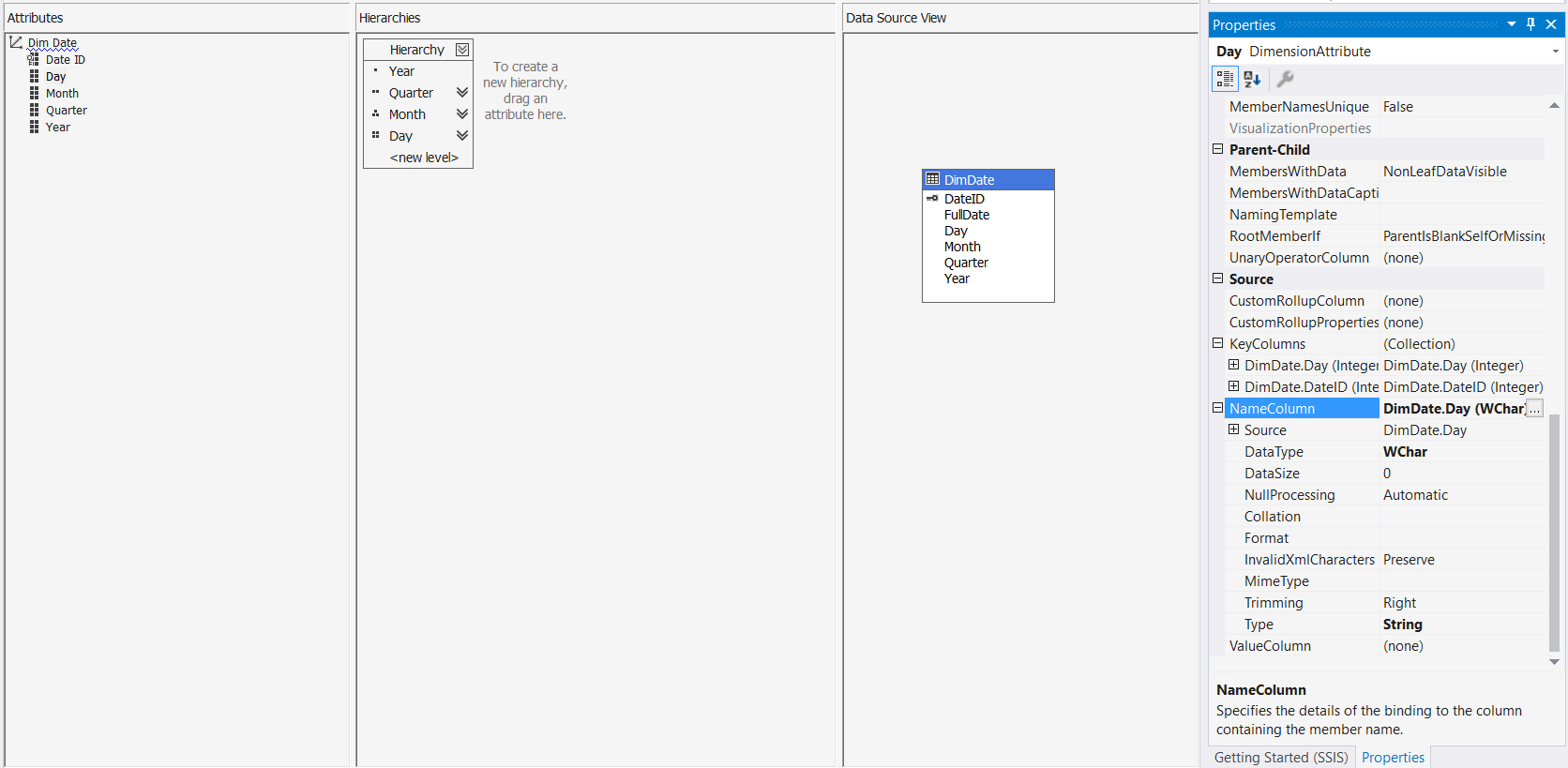


* Xử lý mối quan hệ

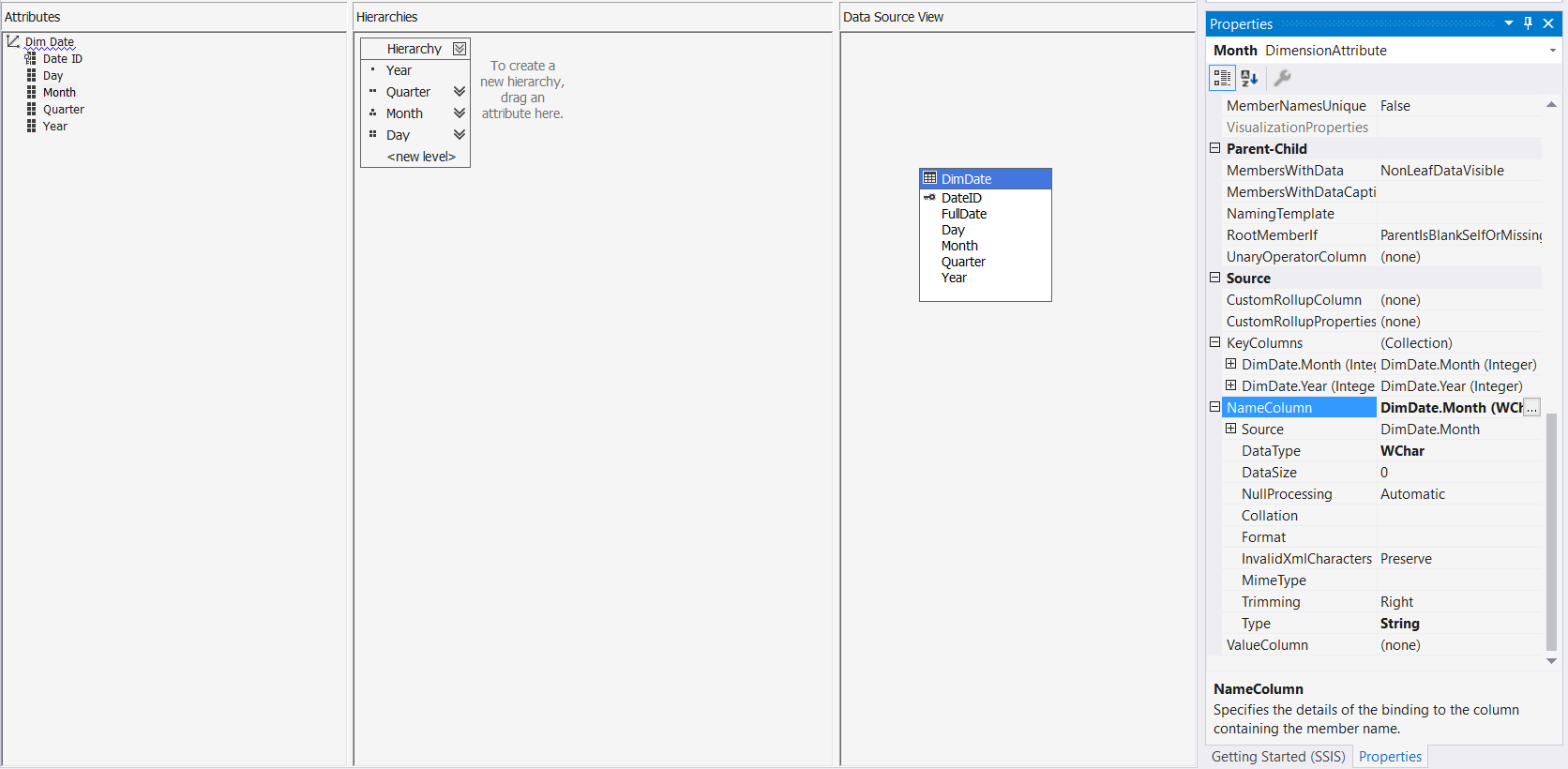


* Xử lý lỗi KeyDuplicate

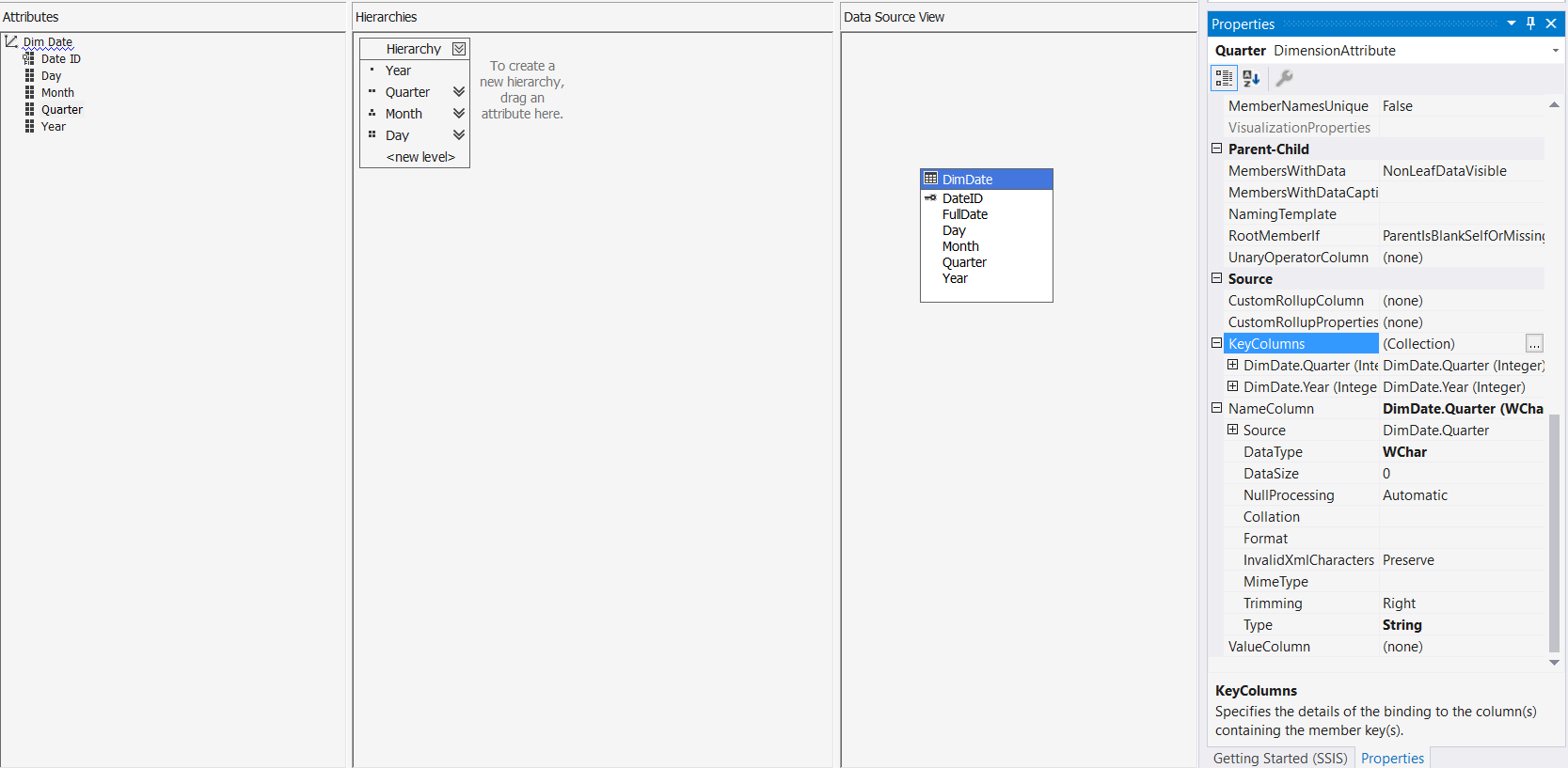
1. Column Day



1. Column Month

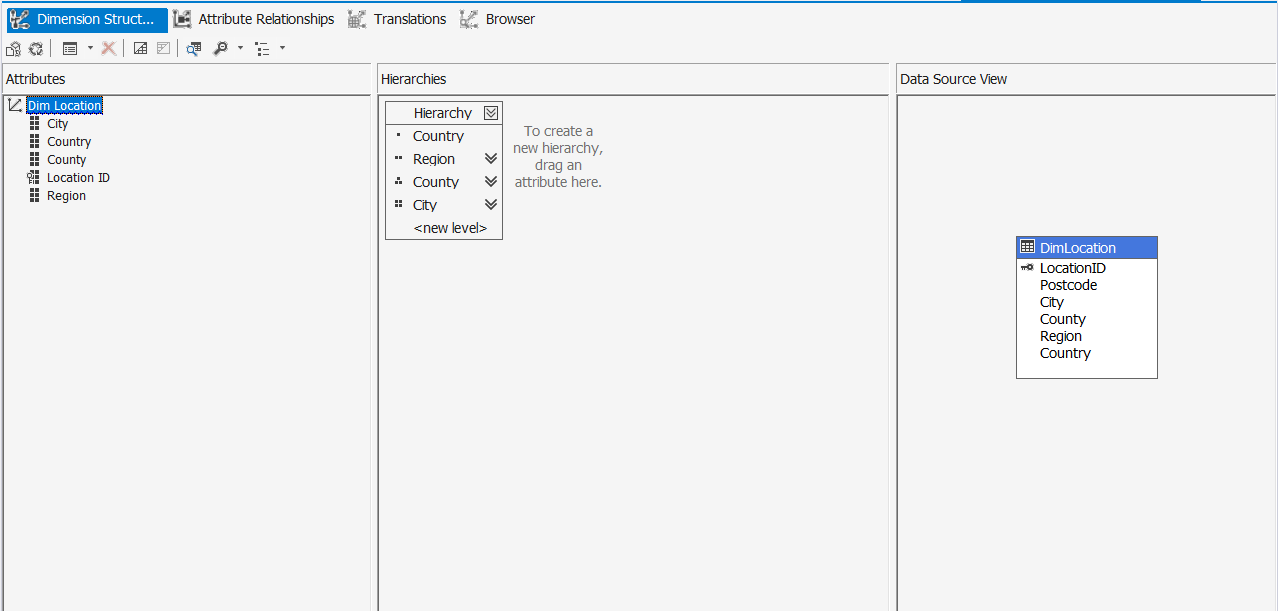


1. Column Quarter

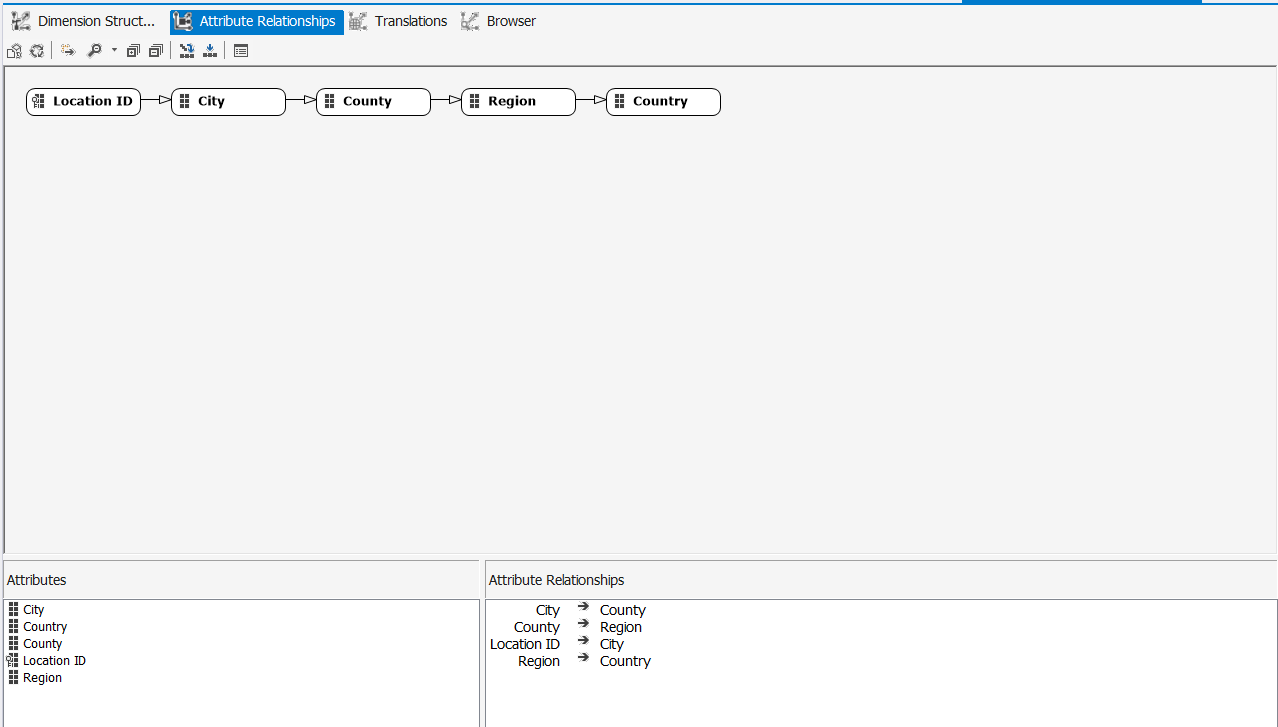


## Tạo phân cấp chiều địa điểm:

1. Tạo bảng kế thừa:



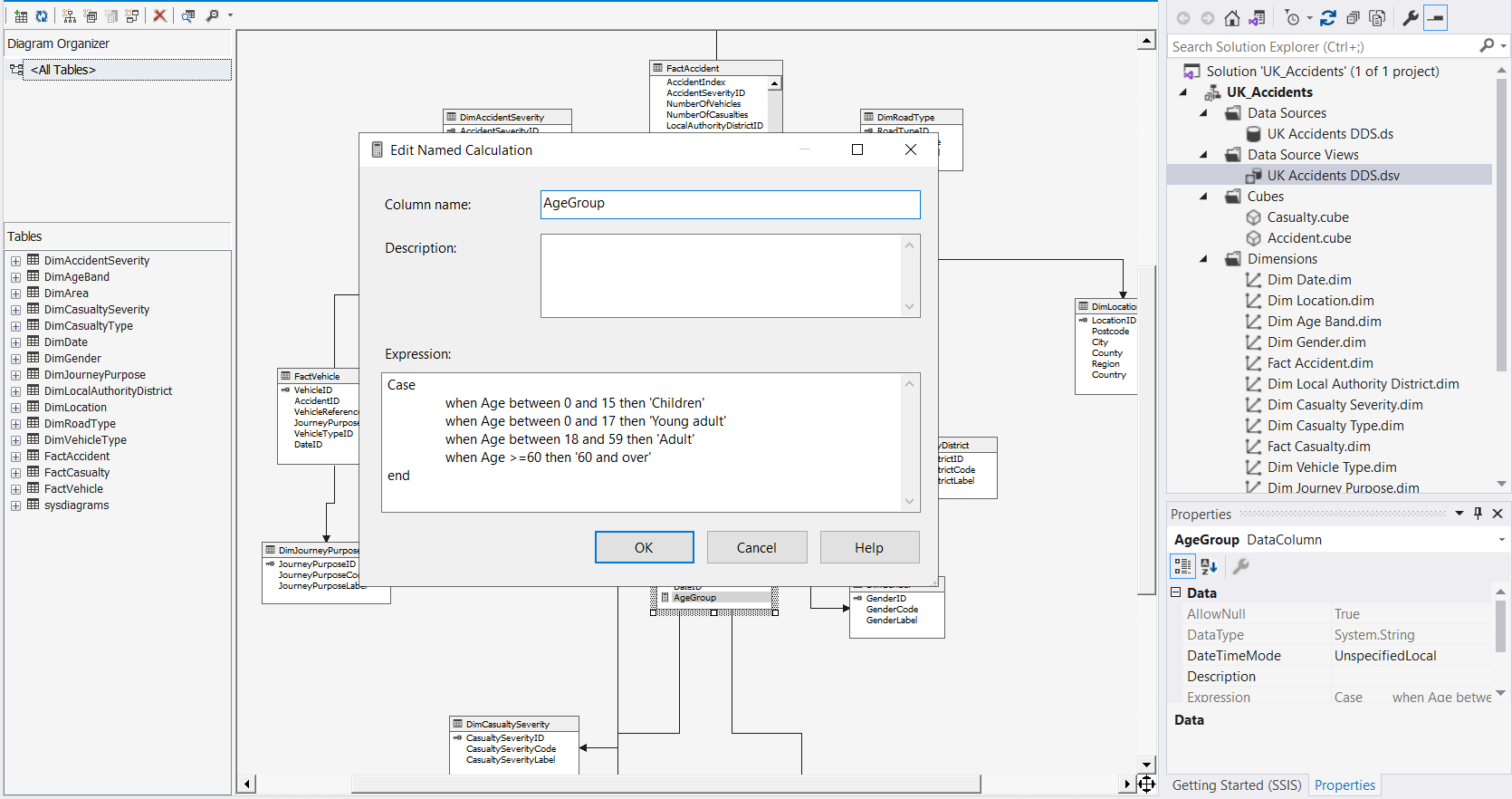
1. Xử lí mối quan hệ



## Tạo thêm các Named Calculation:

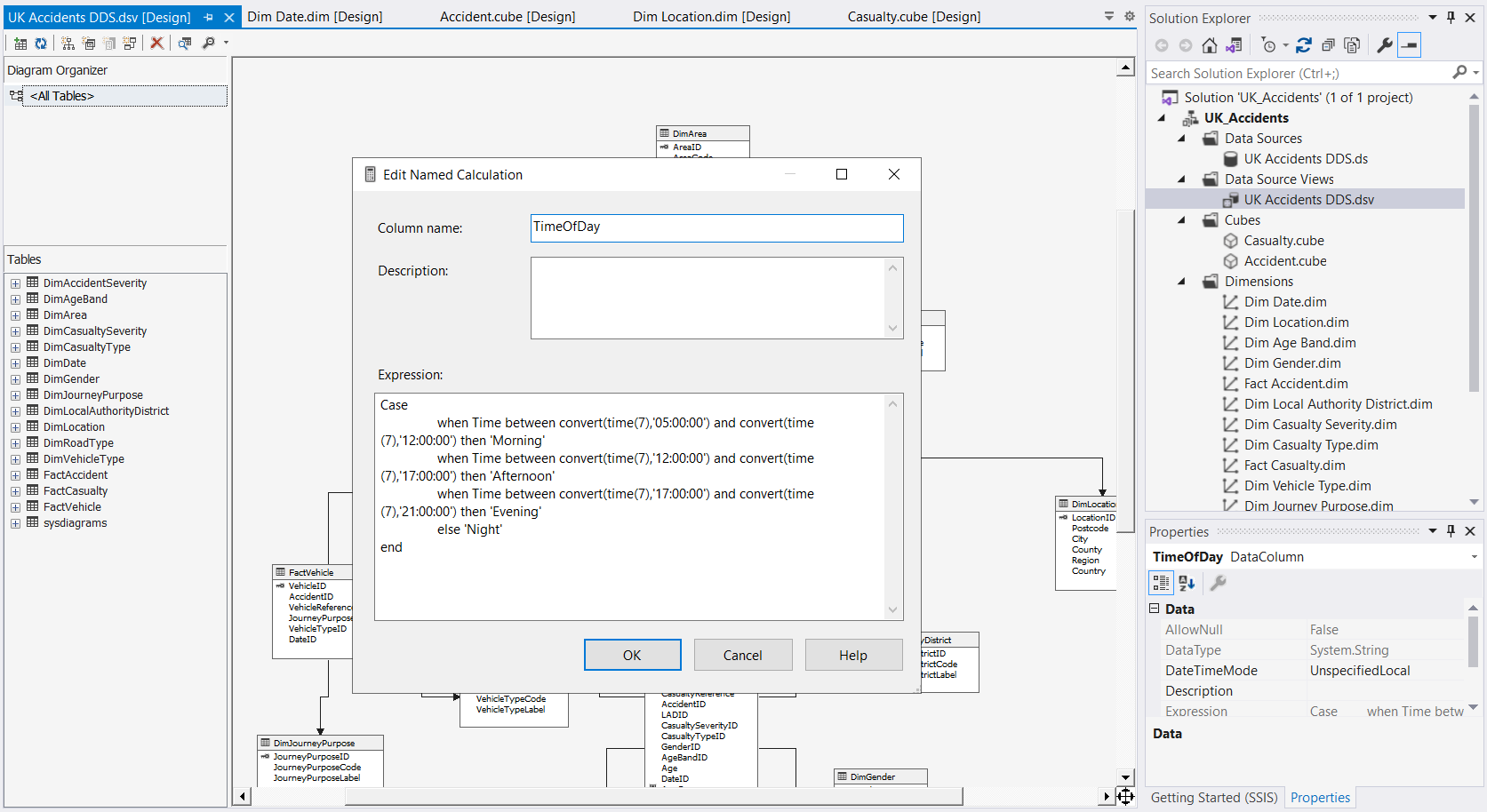
### AgeGroup:

1. Độ Tuổi được định nghĩa như sau:
   1. Children: 0-15
   2. Young adult: 0-17
   3. Adult: 18-59
   4. 60 and over: 60-...

:

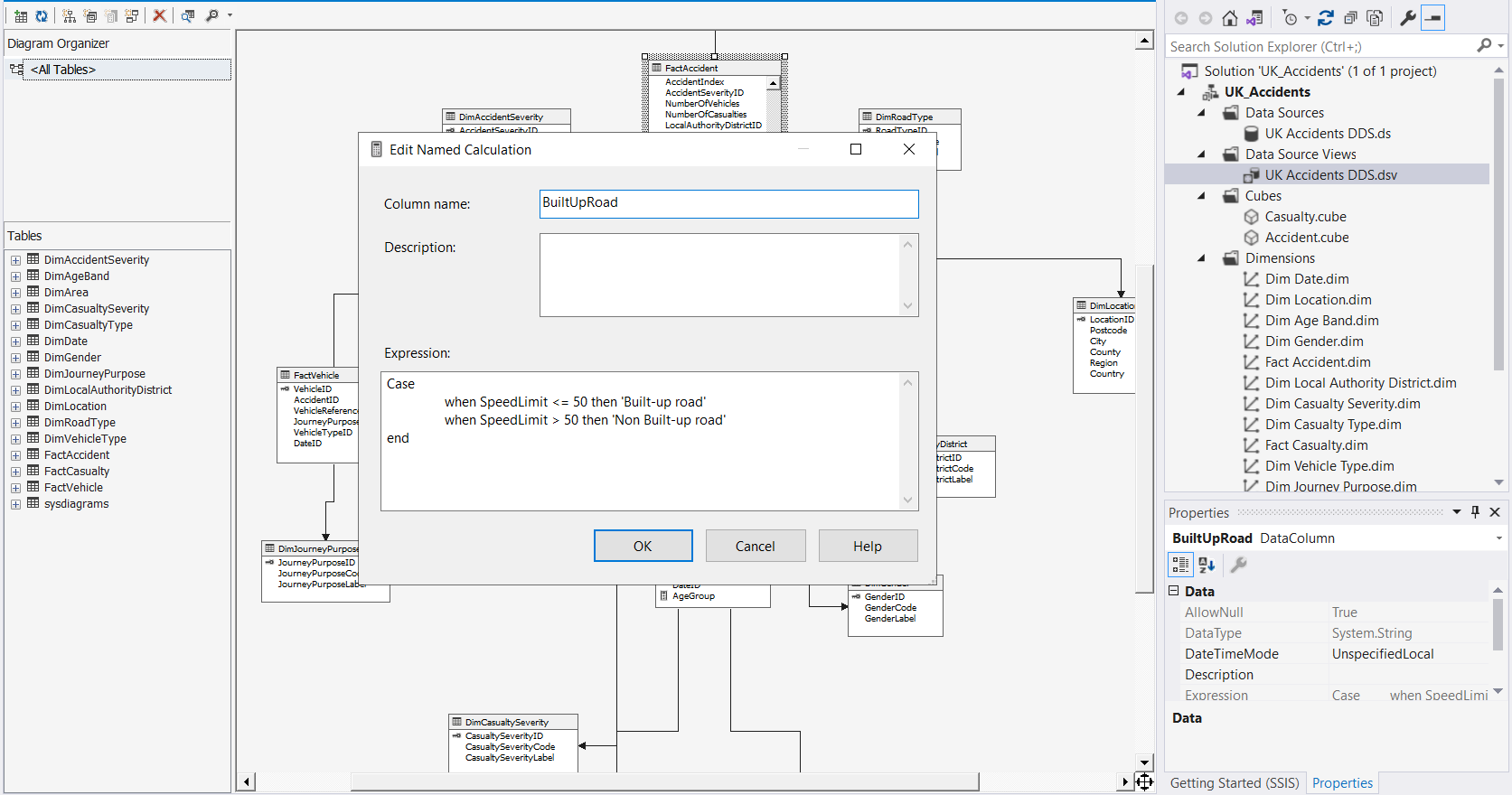
### TimeOfDay:

1. Thời Điểm Trong Ngày
   * Morning: 5am-12pm
   * Afternoon: 12pm-5pm
   * Evening: 5pm-9pm
   * Night: 9pm-5am

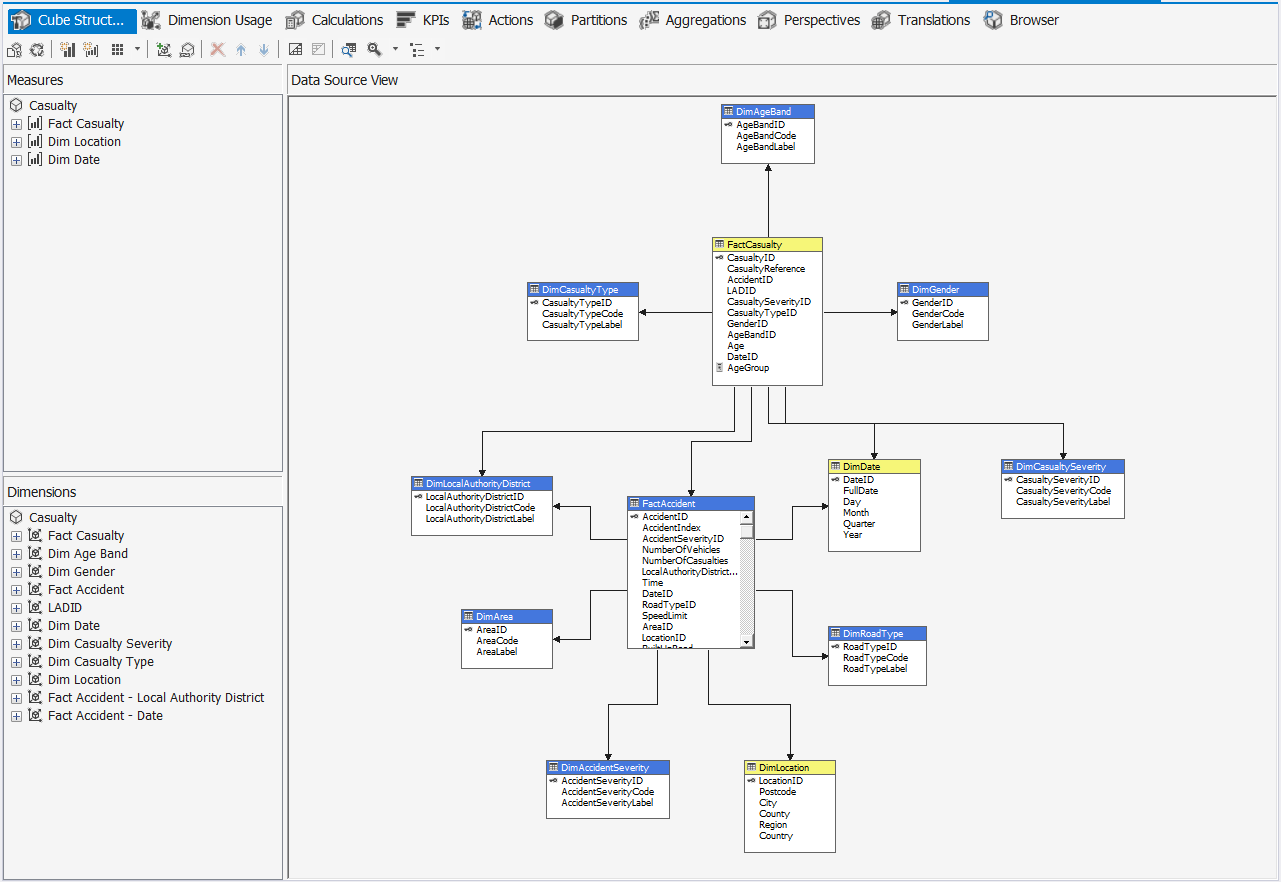


### BuildUpRoad:

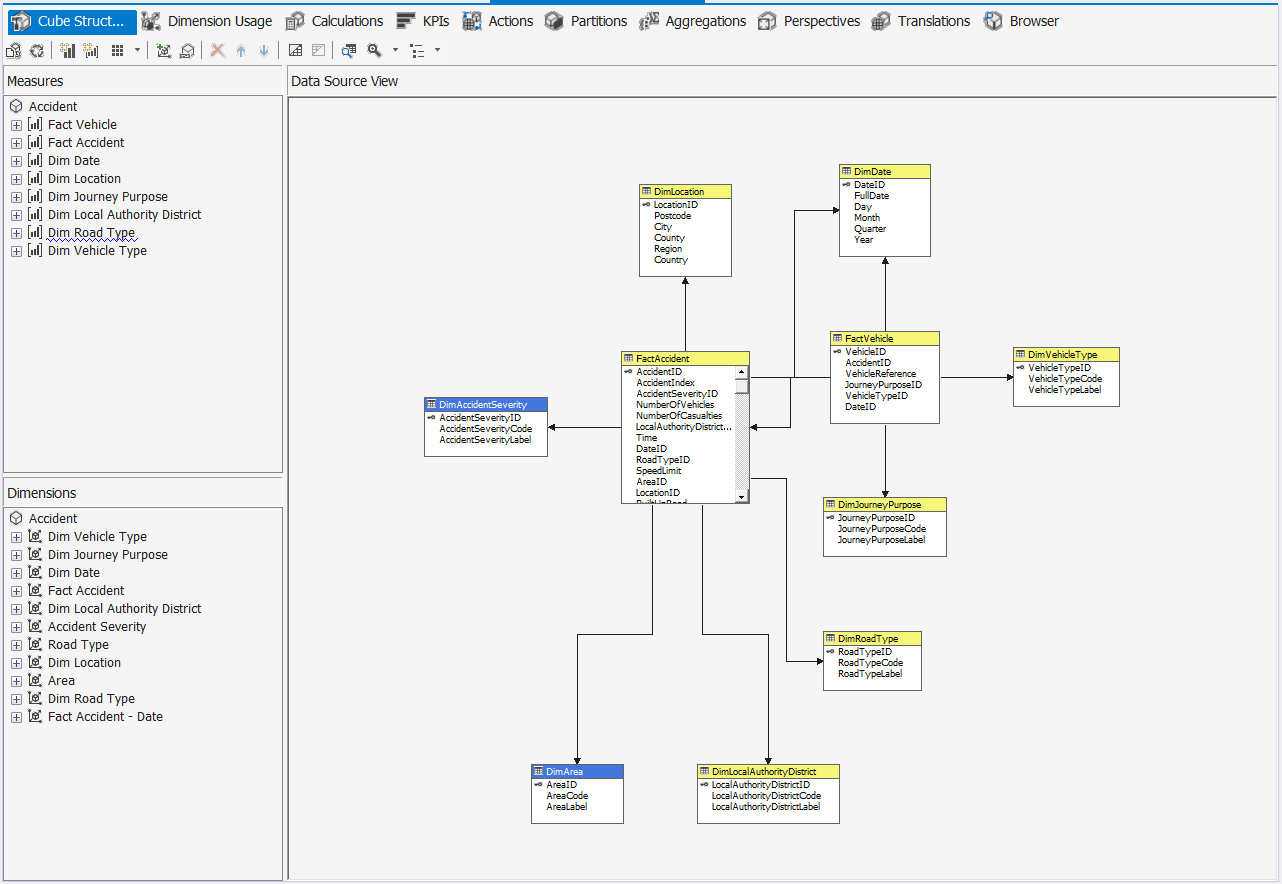
* Built-up Road có 2 giá trị:
  + Built-up road: Nếu tốc độ giới hạn (Speed Limit) dưới 50 mph
  + Non Built-up road: Nếu tốc độ giới hạn từ 50 mph



## Tạo Casualty cube để xem các thống kê các số liệu về thương vong:

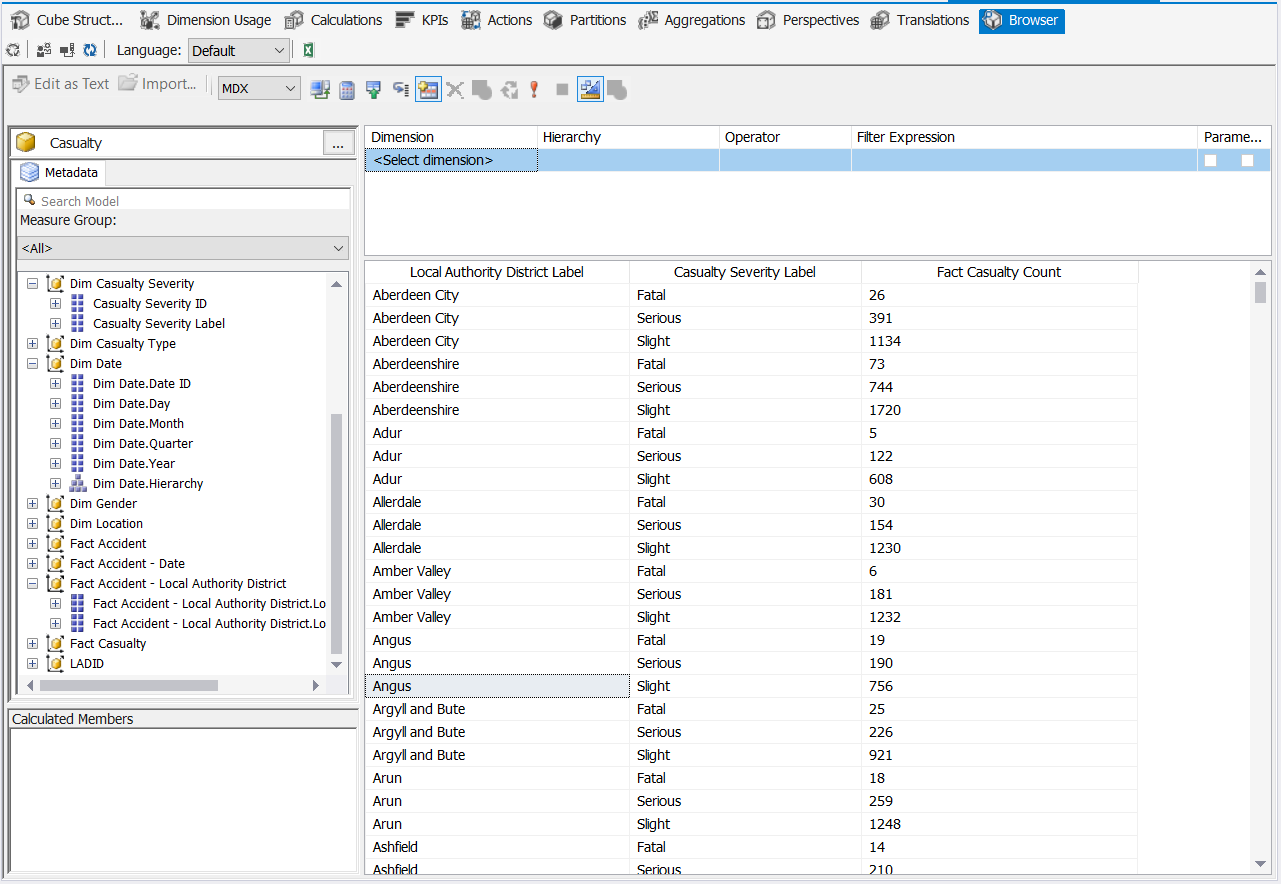


## Tạo Accident cube để xem các thống kê các số liệu về TNGT:

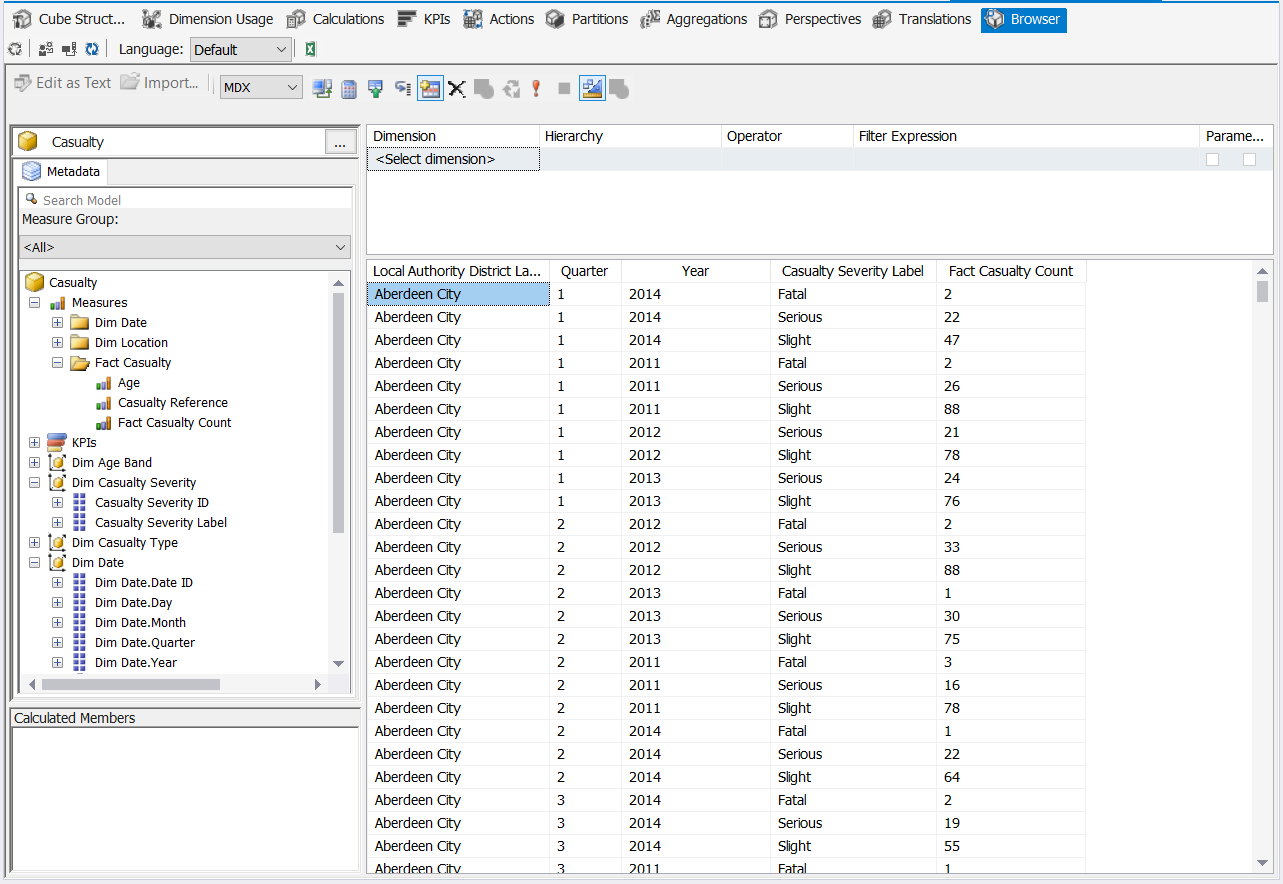


## Thực hiện các thống kê:

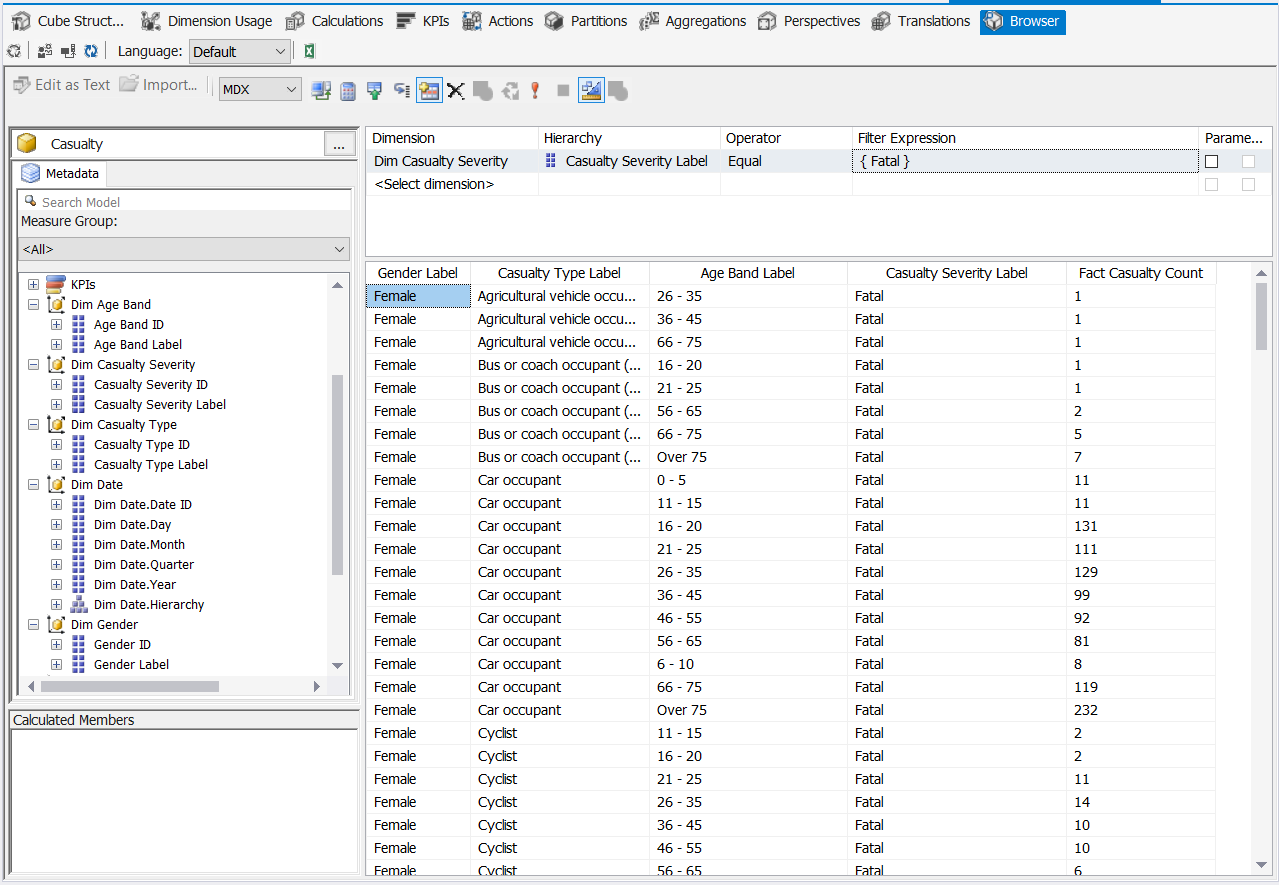
1. Thống kê số lượng nạn nhân theo Mức Độ Nghiêm Trọng (Fatal, Serious, Slight) ở các Địa phương (Local\_Authority\_(District)) trong tất cả các năm.



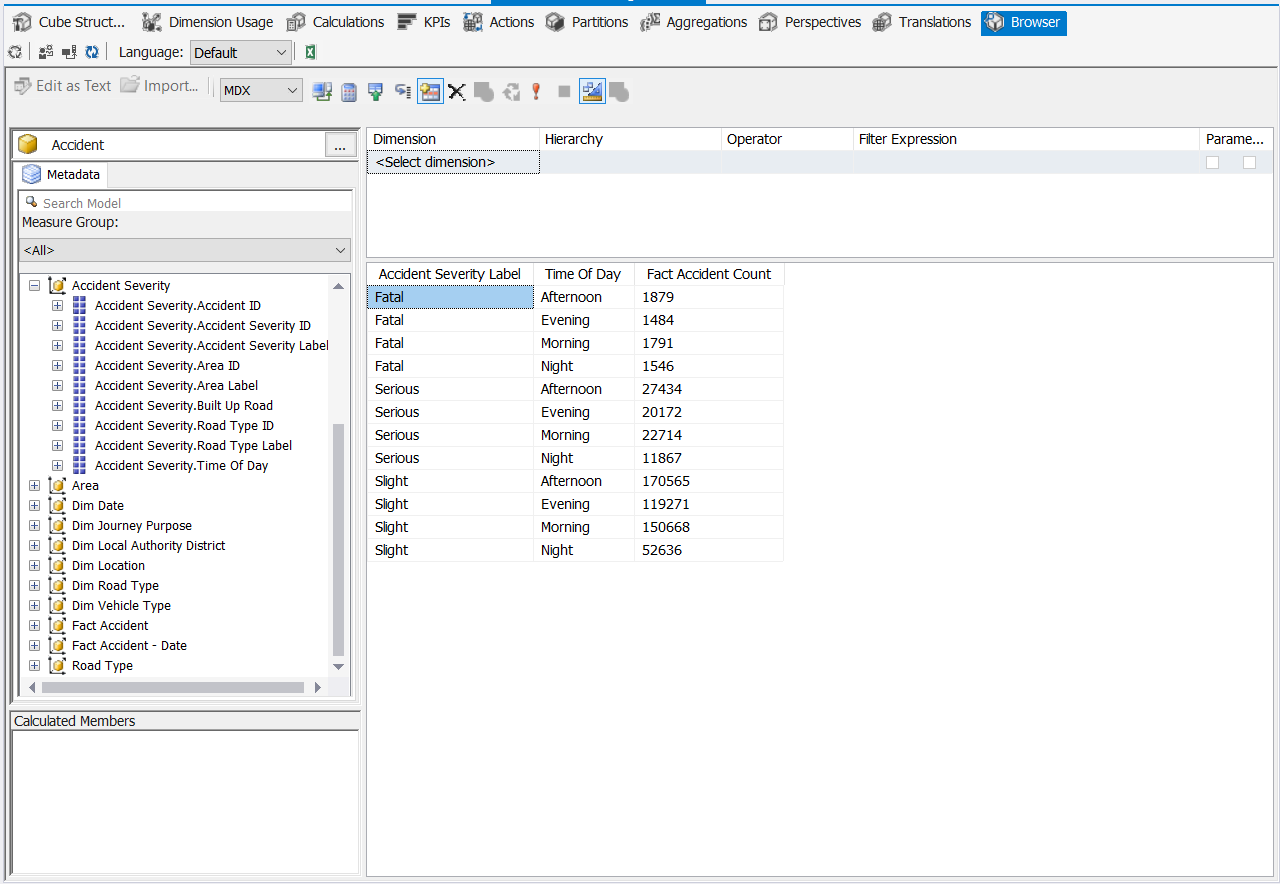
1. Thống kê số lượng nạn nhân theo Mức Độ Nghiêm Trọng ở các Địa Phương (Local\_Authority\_(District)) theo các Quý trong từng năm.



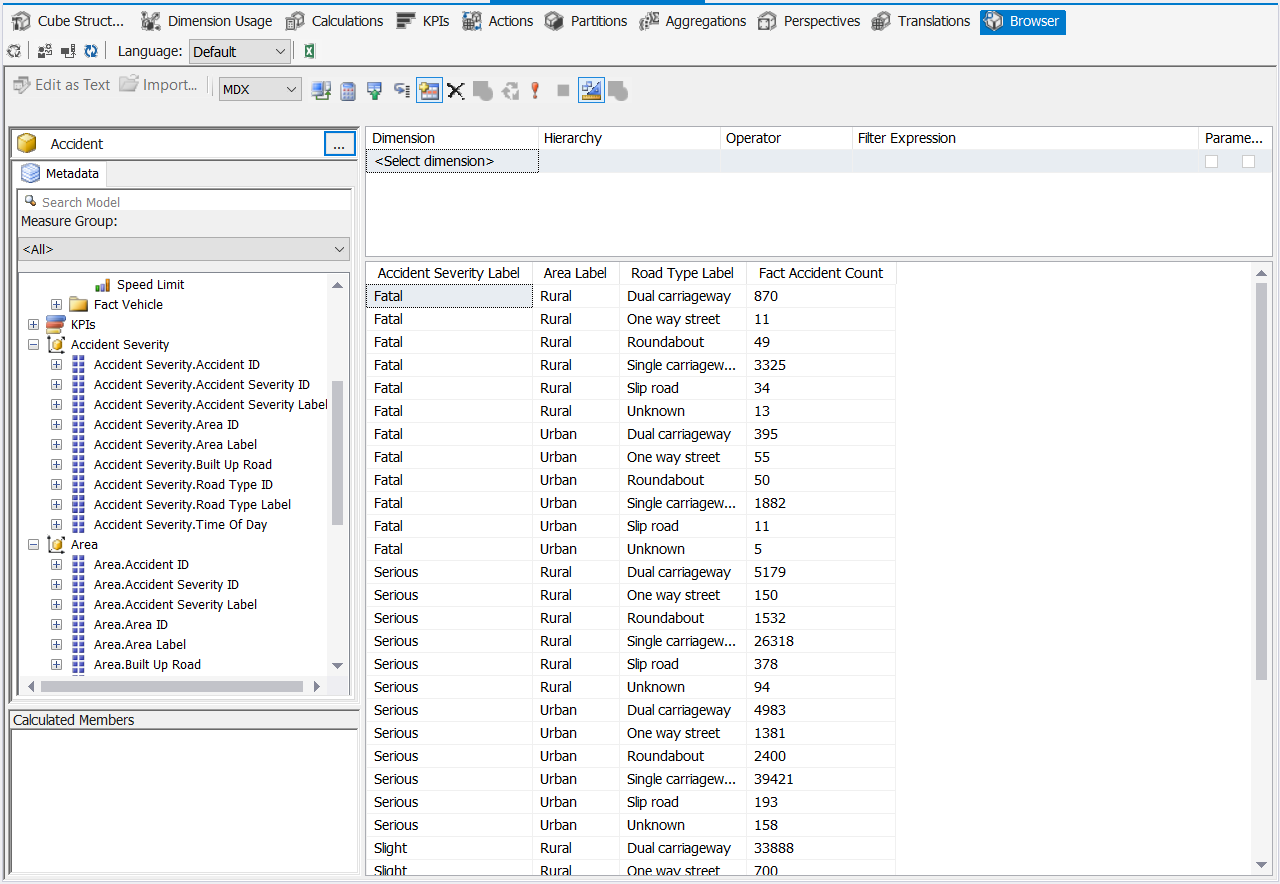
1. Thống kê số lượng người tử vong theo Giới Tính, Loại Nạn Nhân (Casualty Type) và Nhóm Tuổi (Age\_Band\_of\_Casualty) theo các năm.



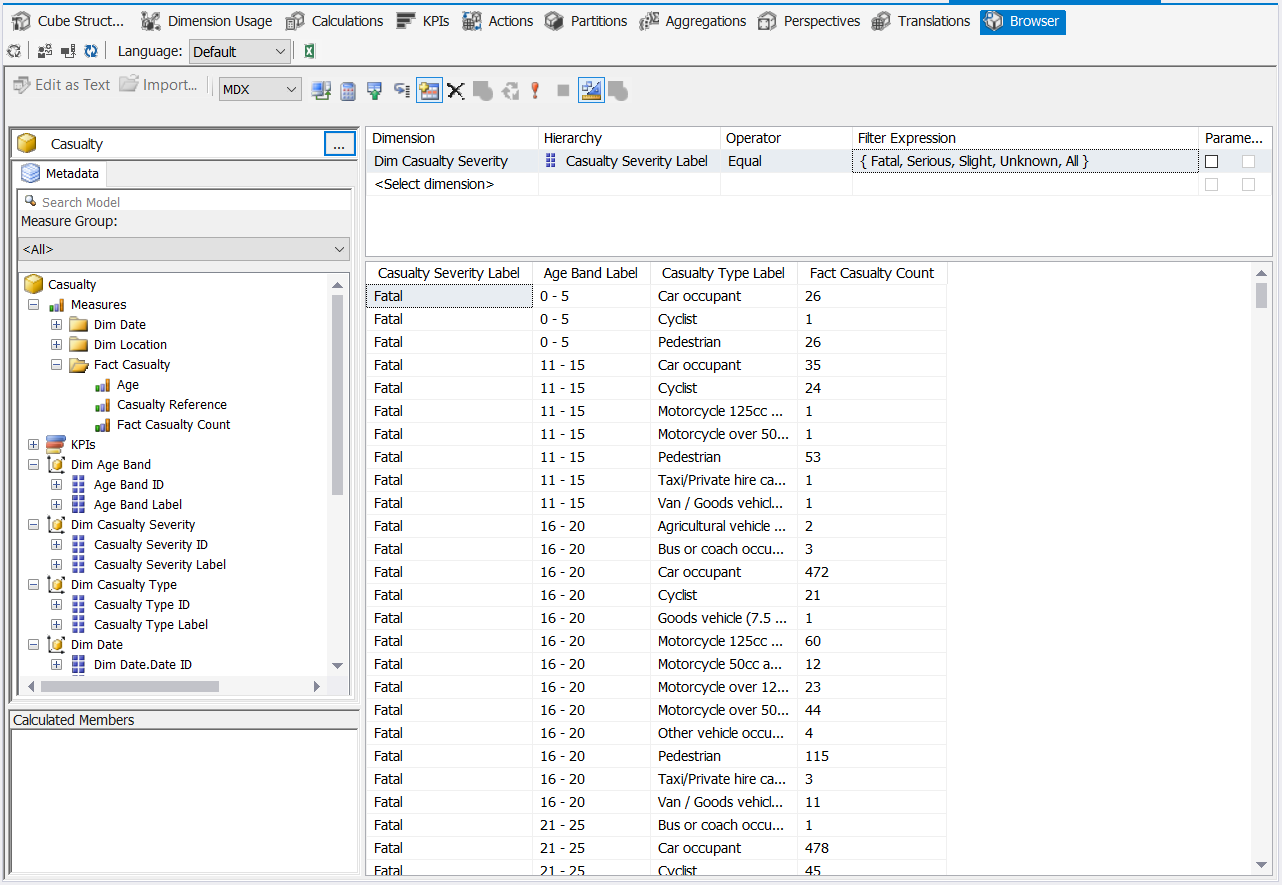
1. Thống kê số lượng TNGT theo Mức Độ Nghiêm Trọng và Thời Điểm Trong Ngày (Morning: 5am-12pm, Afternoon: 12pm-5pm, Evening: 5pm-9pm, Night: 9pm-5am) trong các năm.



1. Thống kê số lượng TNGT theo Mức Độ Nghiêm Trọng, Vùng (Urban\_or\_Rural\_Area), và Kiểu Đường (Road Type) trong các năm.

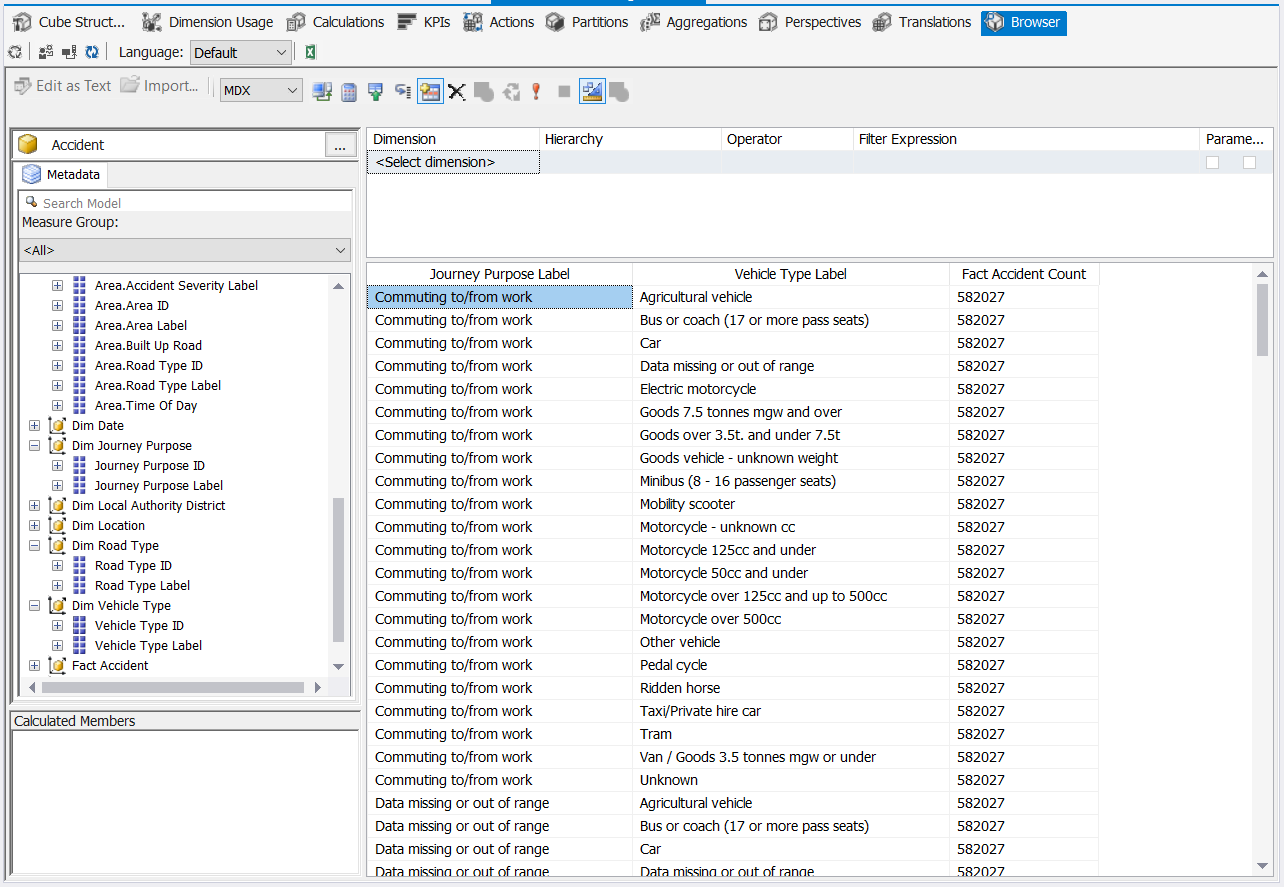


1. Thống kê số lượng nạn nhân theo Mức Độ Nghiêm Trọng, Loại Nạn Nhân (Casualty Type) và Độ Tuổi trong các năm

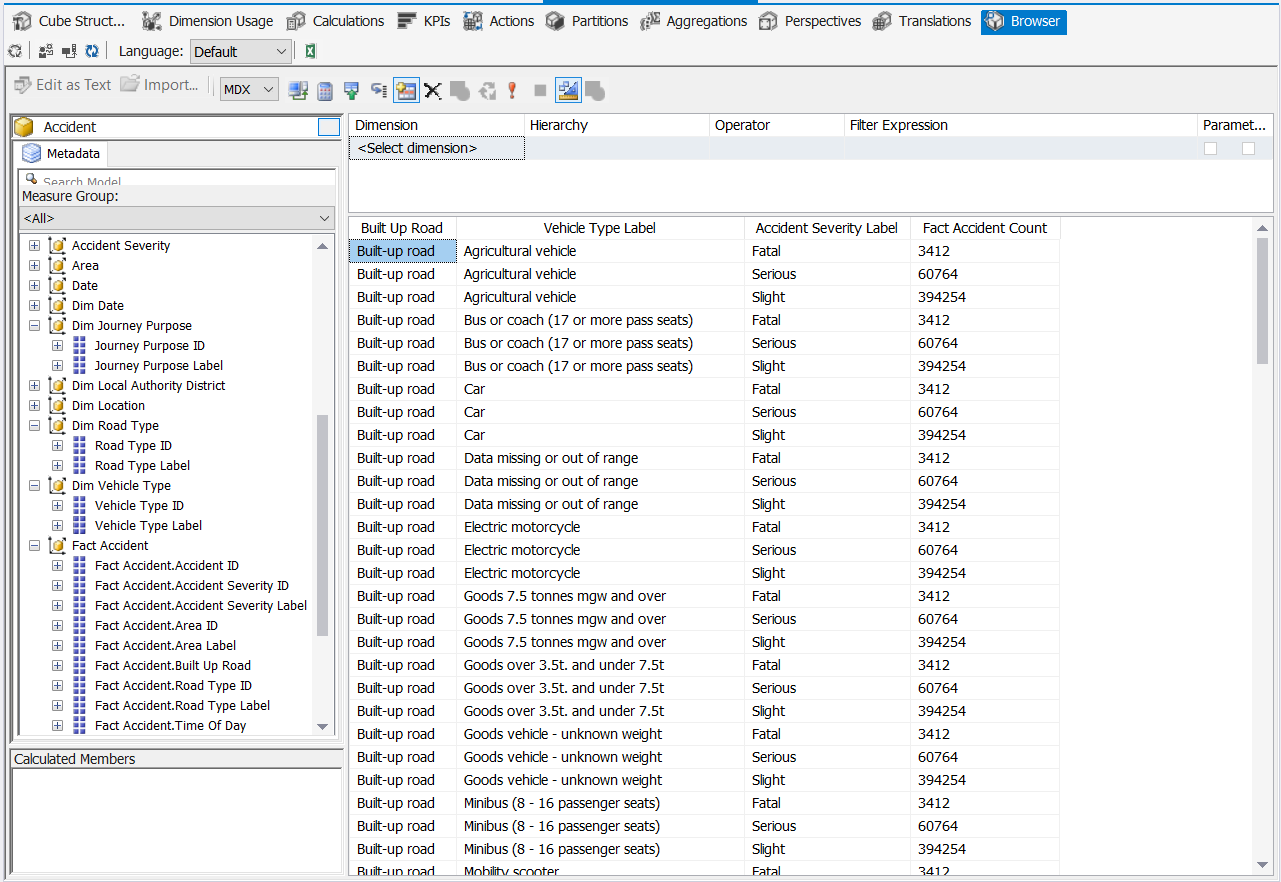


**Tổng hợp**

1. Số lượng tai nạn theo Mục Đích Hành Trình (Journey Purpose) và Loại Phương Tiện (Vehicle\_Type).



1. Thống kê số lượng tai nạn theo Mức Độ Nghiêm Trọng, Loại Phương Tiện (Vehicle Type), Built-up Road trong các năm.



## Một số query MDX cơ bản:

Yêu cầu 1:

Graphical user interface

Description automatically generated with medium confidence

Yêu cầu 2:

A picture containing text

Description automatically generated

Yêu cầu 3:

Table

Description automatically generated

Yêu cầu 4:

Graphical user interface, text, application

Description automatically generated

Yêu cầu 5:

Graphical user interface, application

Description automatically generated

Yêu cầu 6:

Graphical user interface, text

Description automatically generated with medium confidence

Yêu cầu 7:

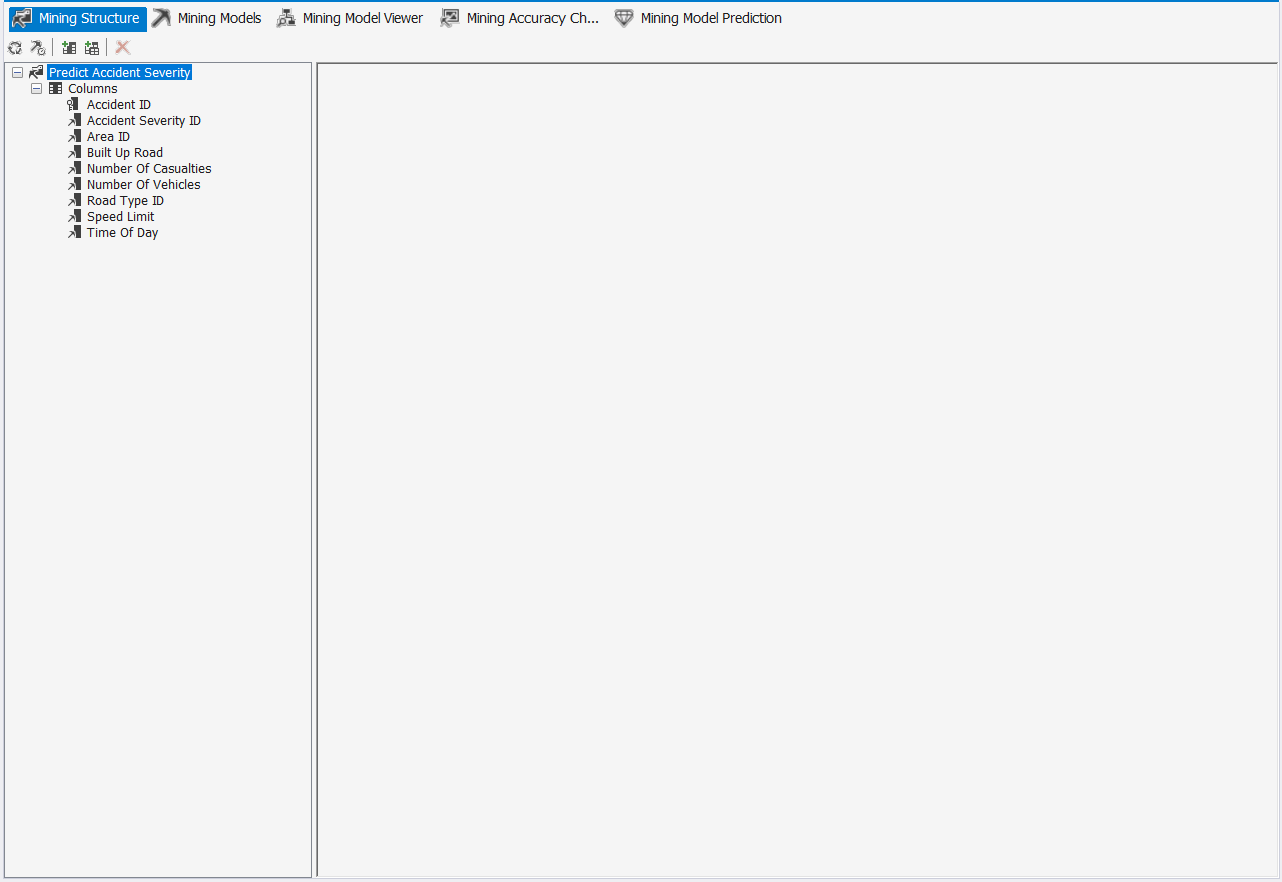
A screenshot of a computer

Description automatically generated

# **THỰC HIỆN MINING:**

## Build Mining Structure:

* Chọn các thuộc tính để làm đầu vào cho mô hình, chọn thuộc tính Accident Severity ID để dự đoán.

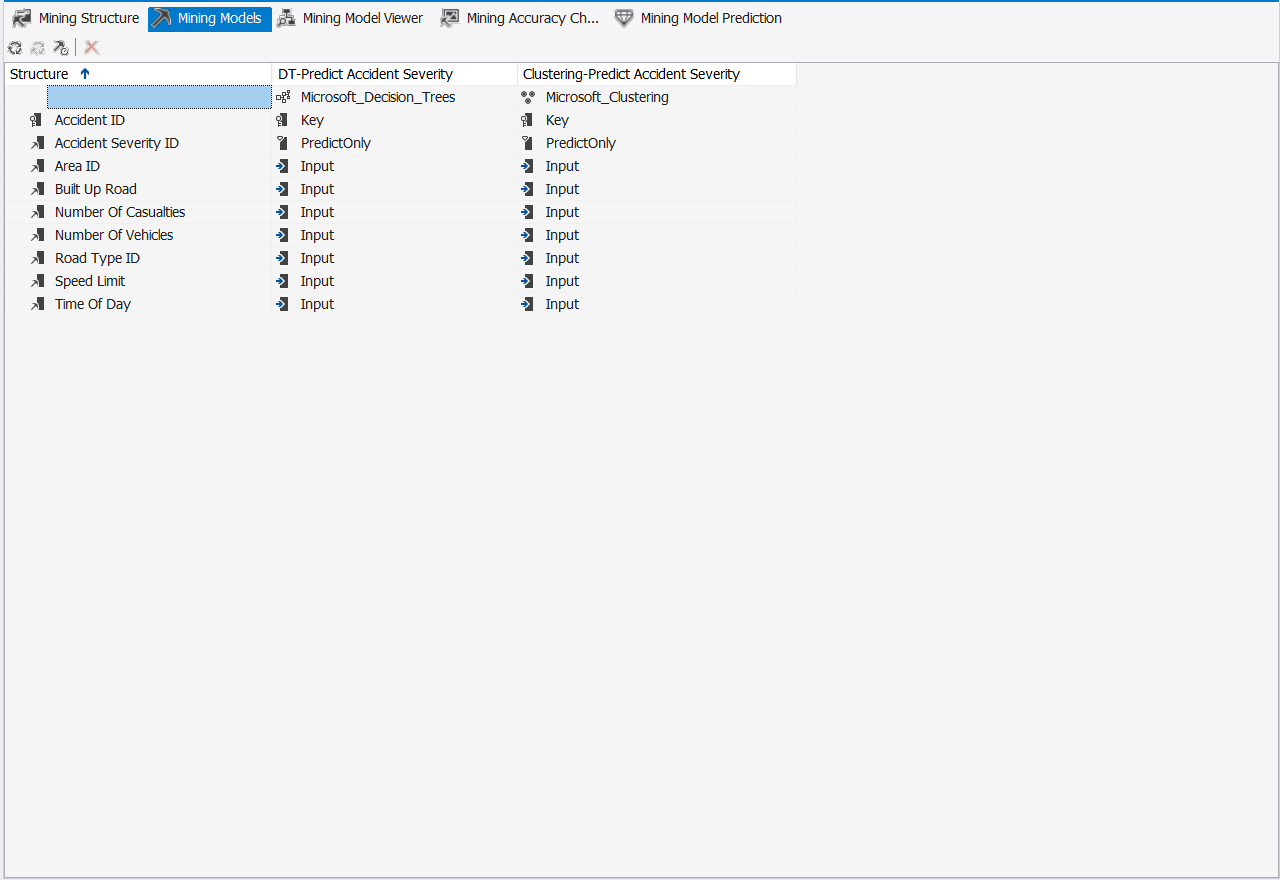


## Chọn Mining Model:

* Lựa chọn 2 mô hình dùng để mining dữ liệu:

+ Cây ra quyết định – Decision Tree

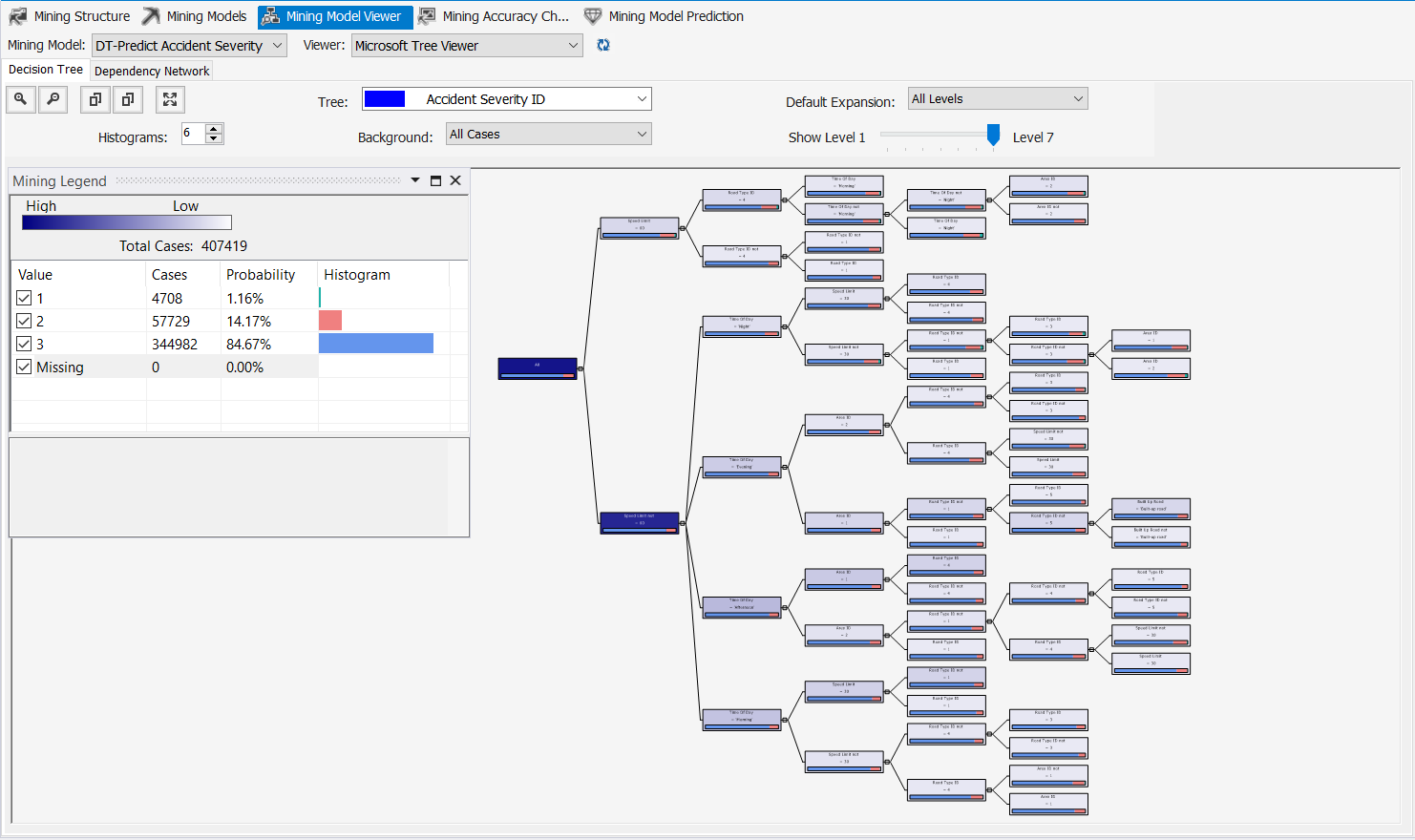
+ Gom nhóm - Clustering



## Xem kết quả Mining:

## Decision Tree:

1. Tổng quan



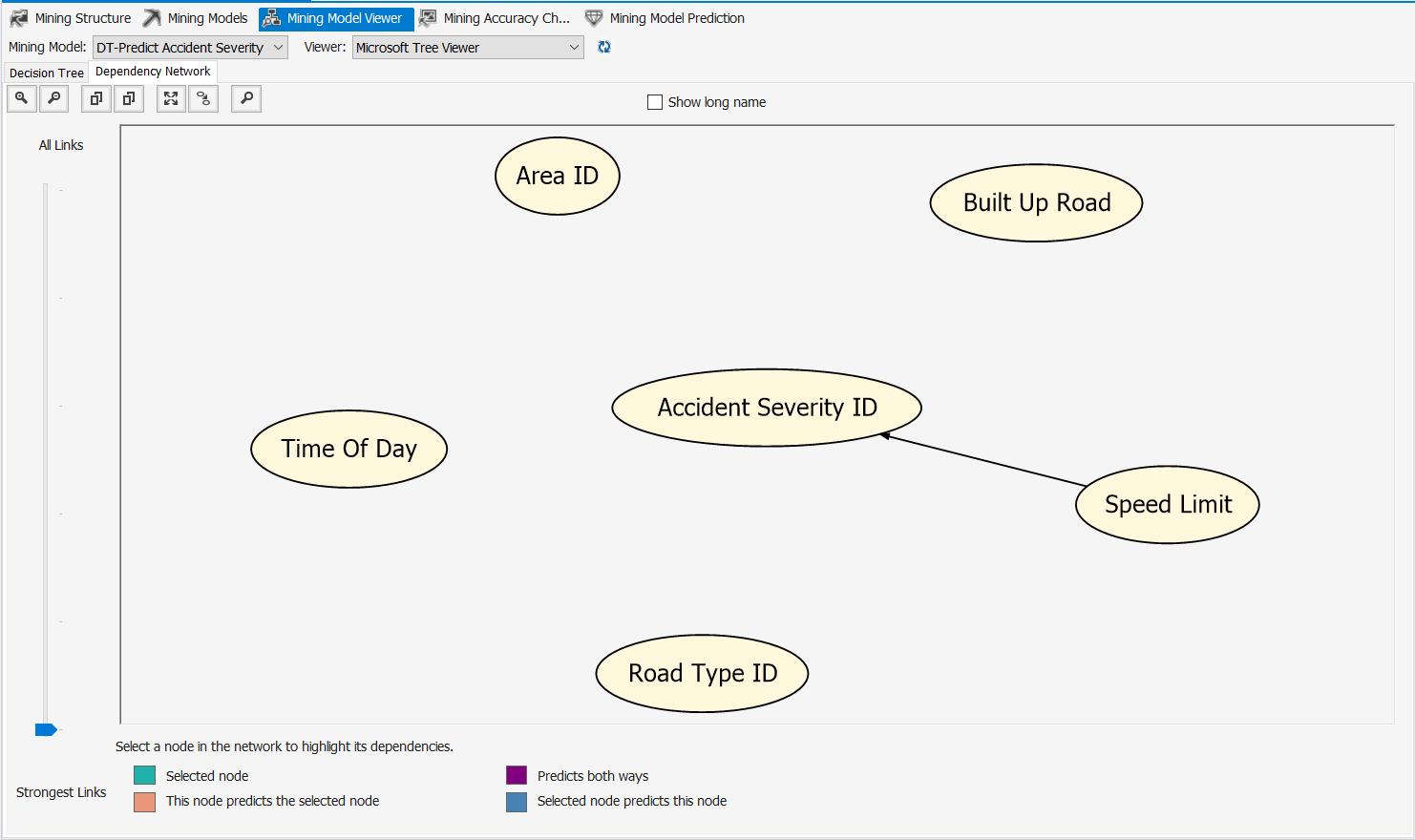
1. Cụ thể

* Với tốc độ không bằng 60 và khoảng thời gian trong ngày rơi vào buổi tối ở khu vực 1 và loại đường khác 1 và 5 thì xác xuất xảy ra tai nạn loại nhẹ sẽ rơi vào khoảng 86%, còn vừa và nặng sẽ có xác xuất thấp hơn.



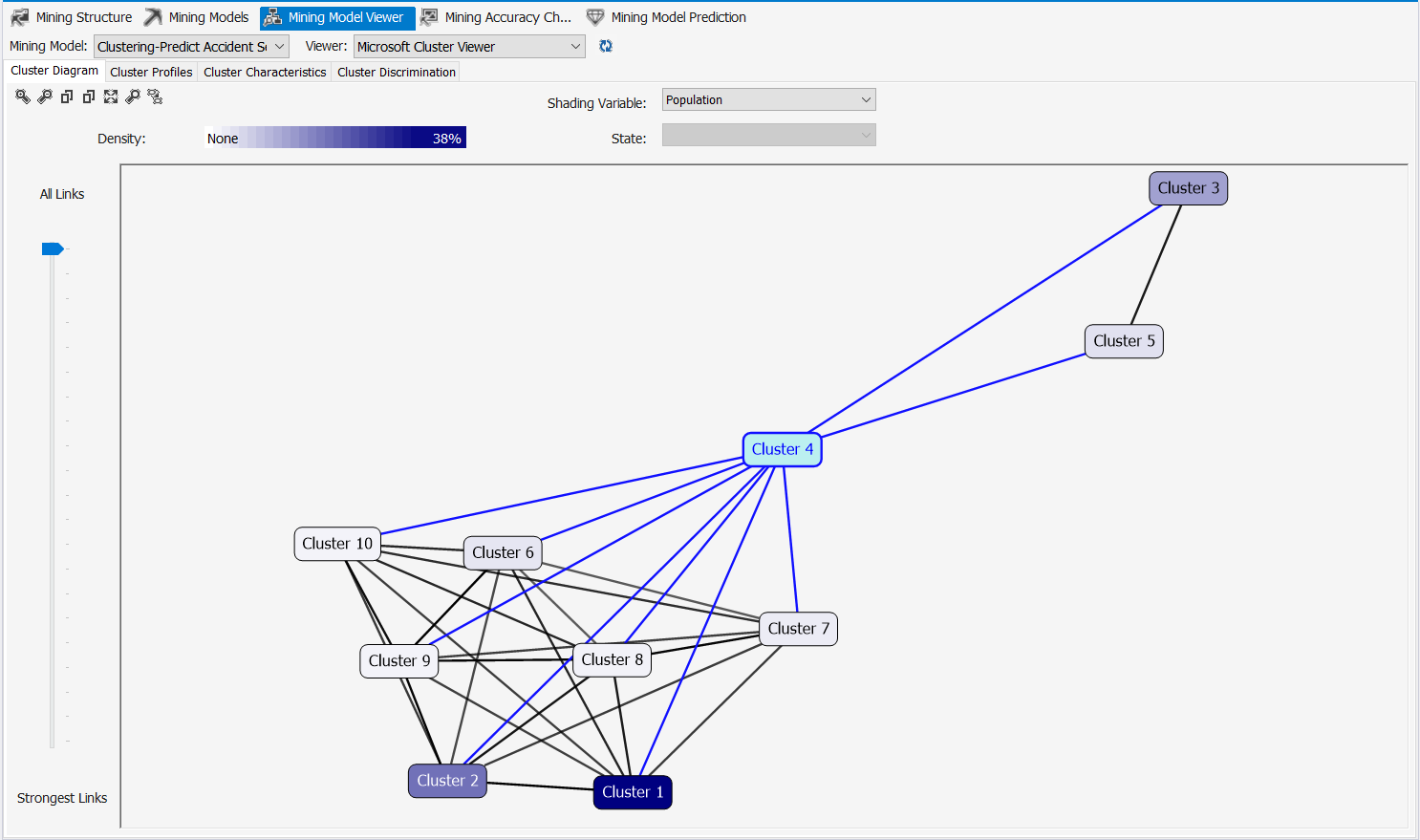
## Dependency Network:

1. Có thể thấy Speed Limit có ảnh hưởng nhiều nhất tới mô hình

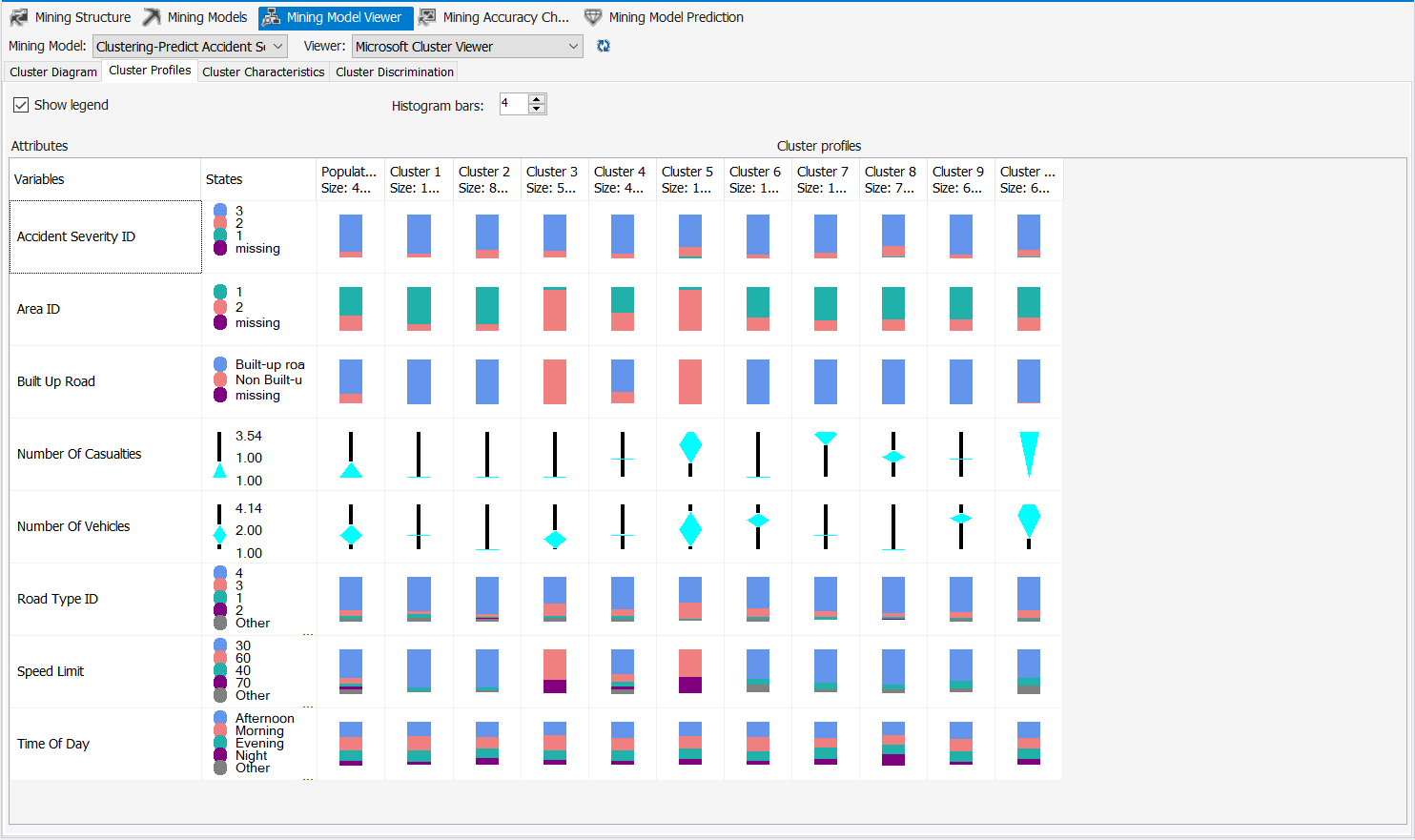


## Clustering:

1. Cluster Diagram

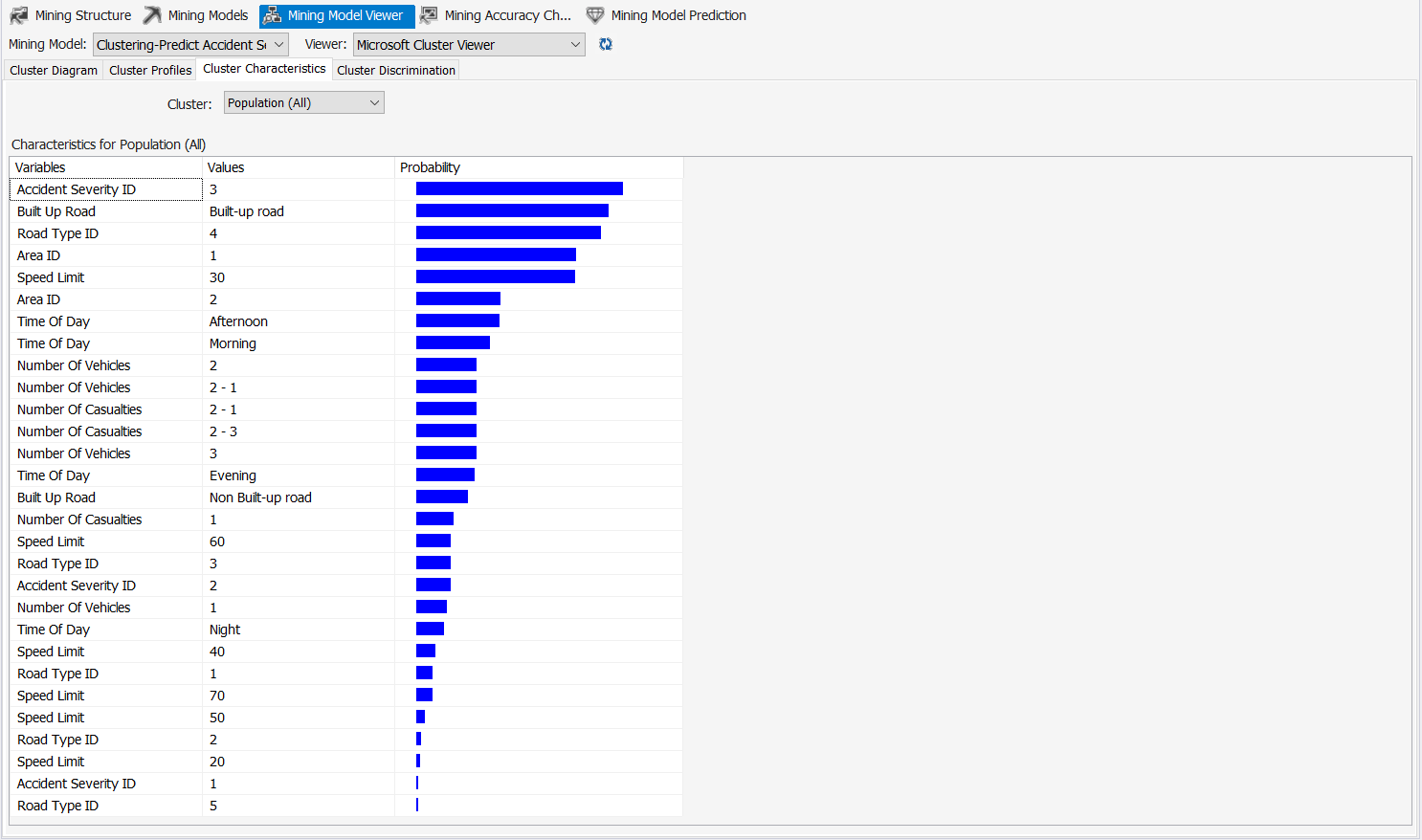


1. Cluster Profile



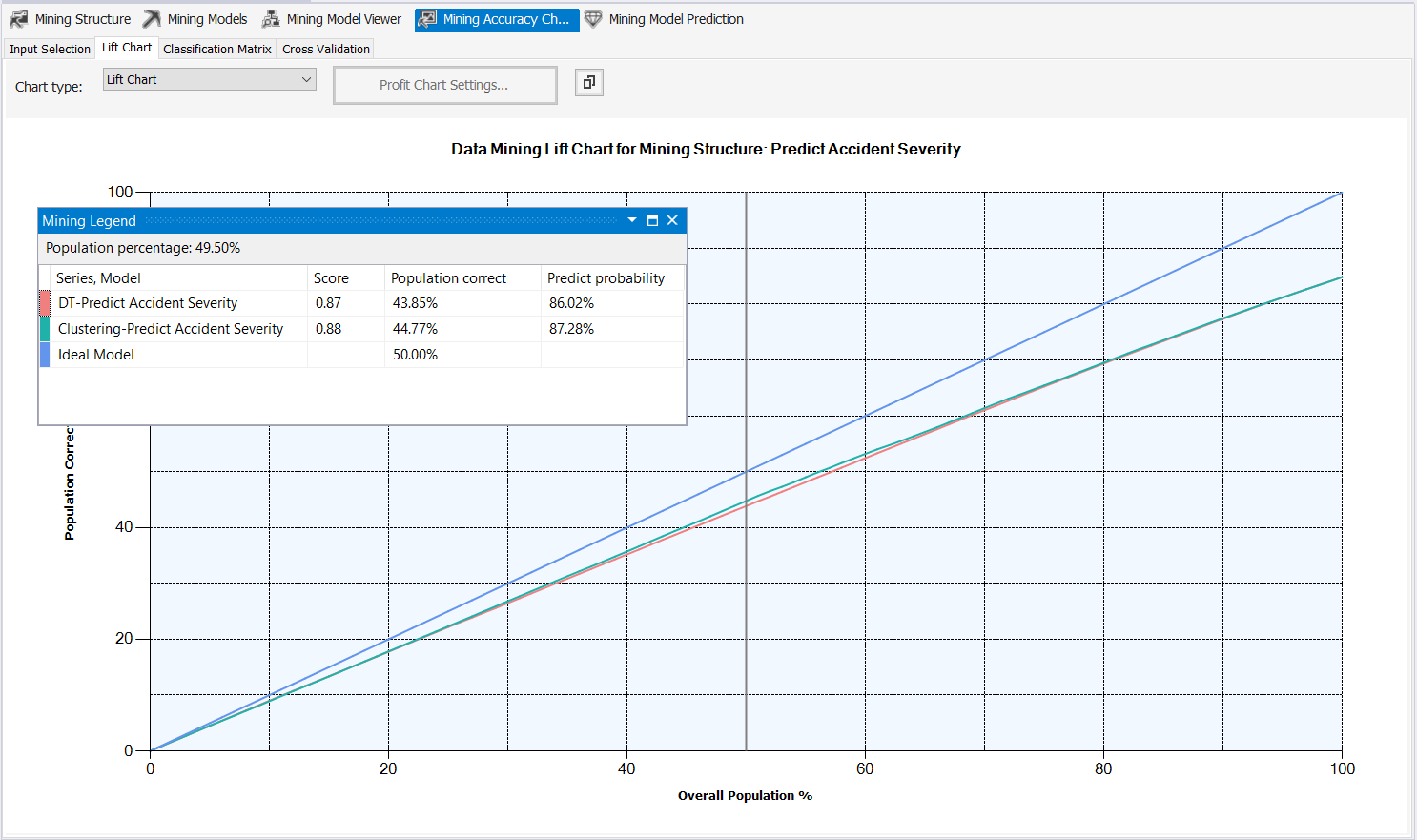
1. Cluster Characteristics

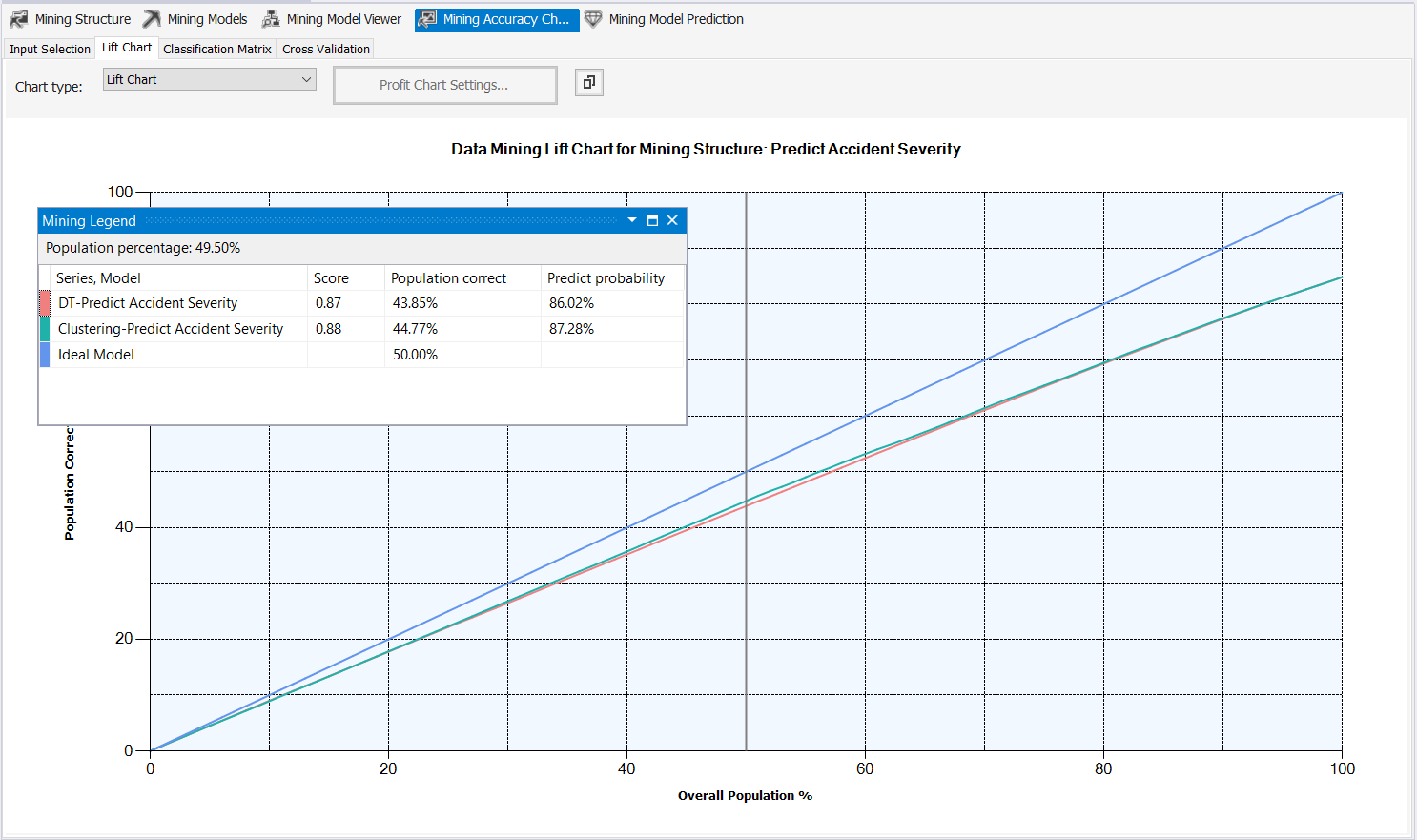
* Các đặc tính nổi bật của mô hình: như là loại tai nạn thường hay xảy ra ở mức độ 3 là mức độ nhẹ và do tốc độ trên 50 km/h,...

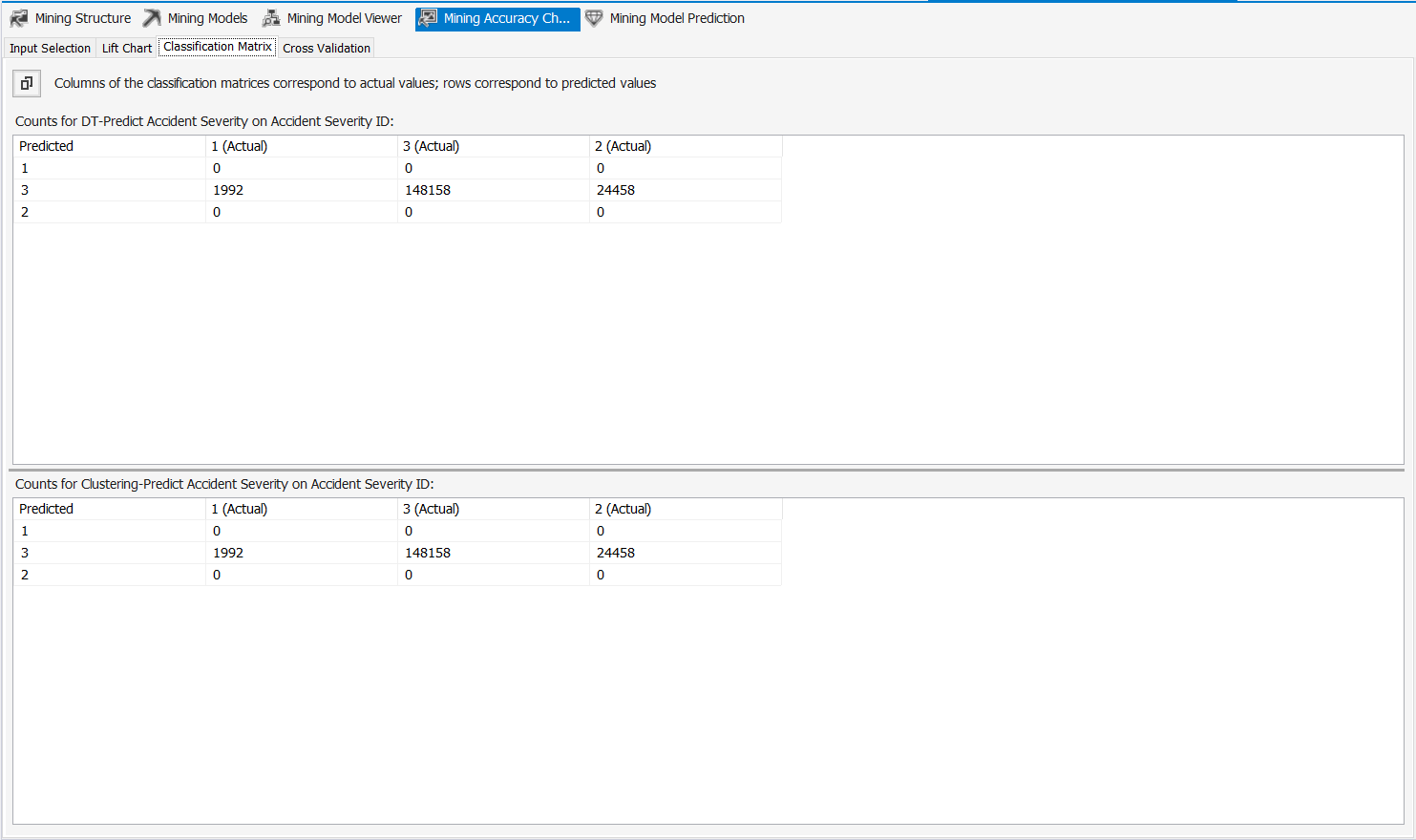


## So sánh:

Nhìn vào kết quả đồ thị, ta có thể thấy 2 mô hình cho kết quả gần như bằng nhau và xấp xỉ 87% so với mô hình lý tưởng



Ma trận phân



# 

# **THỰC HIỆN REPORT BẰNG POWER BI:**

## Câu 1:

**Thống kê số lượng nạn nhân theo Mức Độ Nghiêm Trọng (Fatal, Serious, Slight) ở các Địa phương (Local\_Authority\_(District)) trong tất cả các năm**.

Graphical user interface, text, application, chat or text message

Description automatically generated

Table

Description automatically generated

A picture containing bar chart

Description automatically generated

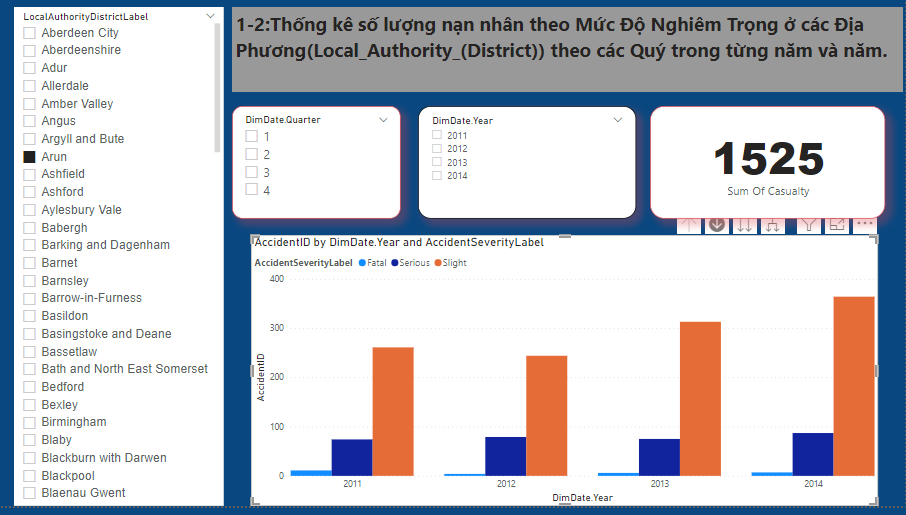
**Nhận xét:**

* + Trong 4 năm từ 2011 đến 2014, số lượng nạn nhân bị tai nạn nhẹ rất nhiều và nhiều hơn gấp 54 lần mức độ tai nạn chết người, và gấp 6 lần tai nạn nặng.
  + Số lượng nạn nhân có xu hứơng giảm dần theo các năm nhưng đến 2014 thì lại tang ở 2 mức độ nặng và nhẹ.
  + Birmingham là địa phương có số lượng nạn nhân bị thương nhẹ và nặng nhiều nhất và nhiều gấp 2-4 lần các chỗ khác

1. **Thống kê số lượng nạn nhân theo Mức Độ Nghiêm Trọng ở các Địa Phương(Local\_Authority\_(District)) theo các Quý trong từng năm**.

Table

Description automatically generated



Chart, bar chart

Description automatically generated

**Nhận xét:**

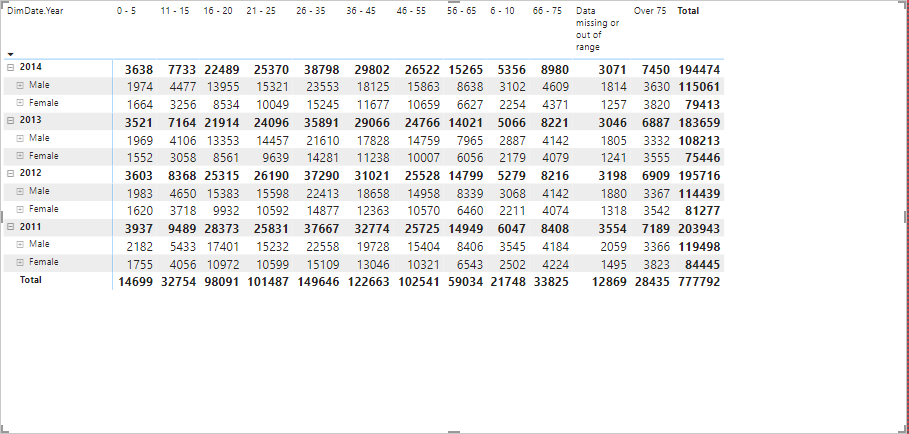
* + Số lượng nạn nhân các quý trong các năm không thay dổi quá nhiều
  + Birmingham là địa phương có số lượng nạn nhân bị thương nhẹ và nặng nhiều nhất và nhiều gấp 2-4 lần các chỗ khác trong tất cả các quý

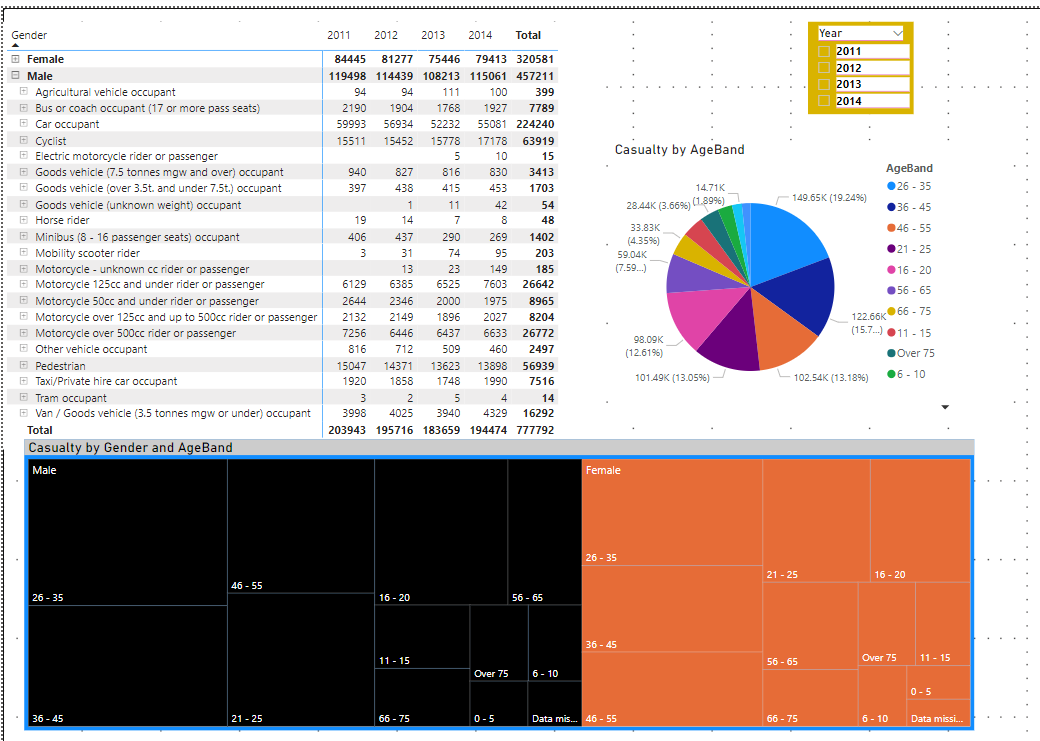
## Câu 2:

**Thống kê số lượng người tử vong theo Giới Tính, Loại Nạn Nhân (Casualty Type) và Nhóm Tuổi (Age\_Band\_of\_Casualty) theo các năm.**

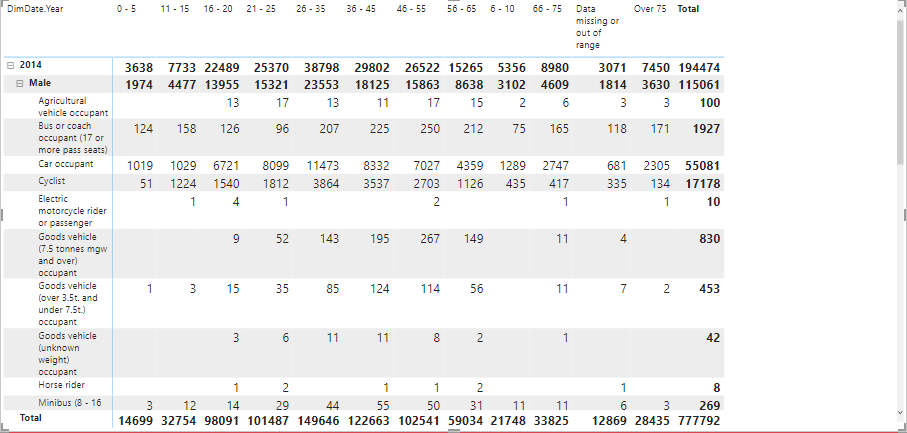
**Bảng tham khảo:**

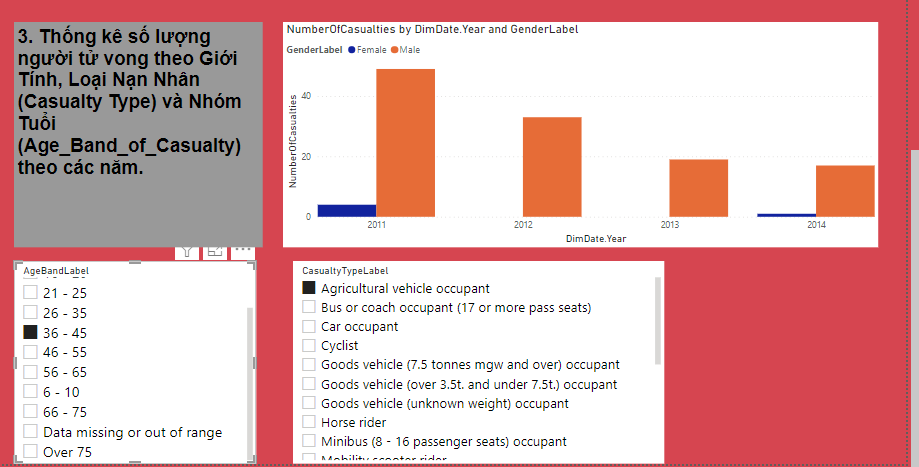
* **Cách 1:**





* **Cách 2:**





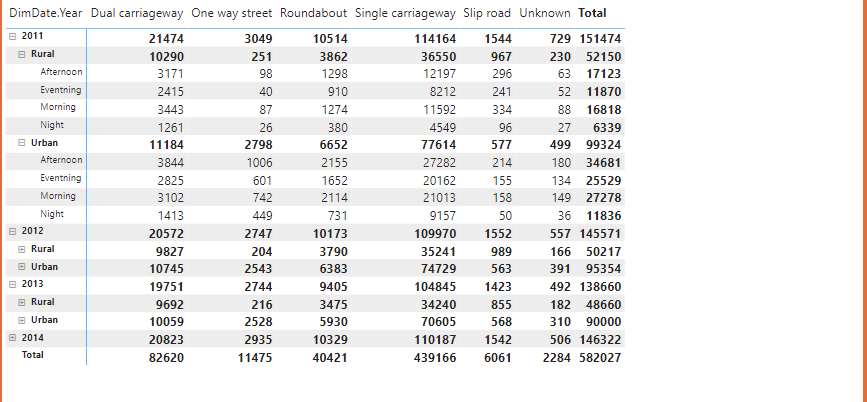
* **Nhận xét:**

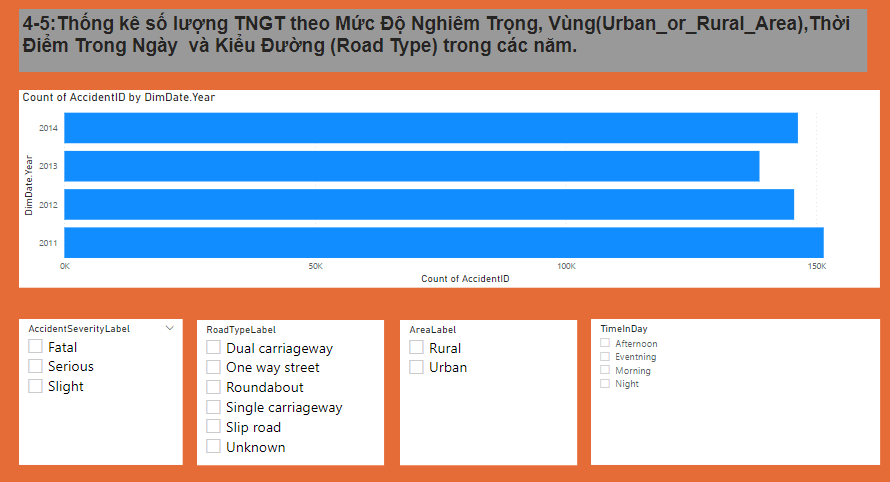
## Câu 4:

**Thống kê số lượng TNGT theo Mức Độ Nghiêm Trọng và Thời Điểm TrongNgày (Morning: 5am-12pm, Afternoon: 12pm-5pm, Evening: 5pm-9pm, Night: 9pm-5am) trong các năm. (LÀM CÙNG CÂU 5)**

## Câu 5

**Thống kê số lượng TNGT theo Mức Độ Nghiêm Trọng, Vùng (Urban\_or\_Rural\_Area), và Kiểu Đường (Road Type) trong các năm.**





**Nhận xét:**

* Số lượng TNGT năm 2013 là thấp nhất về cả tổng các thời điểm trong ngày và từng thời điểm
* Số lượng TNGT các năm của loại đường single carriageway chiếm nhiều nhất và gấp rất nhiều lần các loại khác, trong tất cả năm
* Số lượng TNGT các năm của Urban luôn nhiều hơn Rural

## Câu 6:

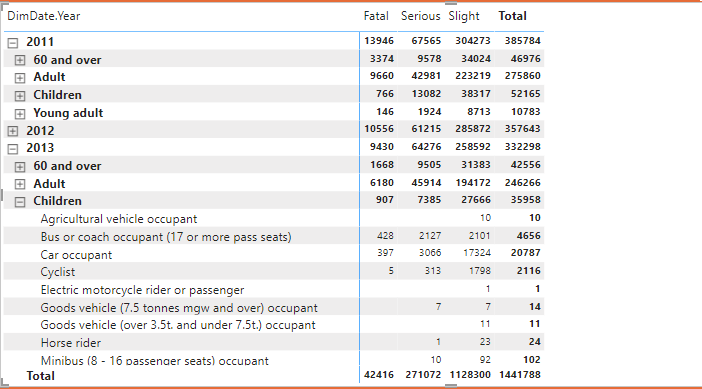
**Thống kê số lượng nạn nhân theo Mức Độ Nghiêm Trọng, Loại Nạn Nhân (Casualty Type) và Độ Tuổi trong các năm, Độ Tuổi được định nghĩa như sau:**

Children: 0-15

Young adult: 0-17

Adult: 18-59

1. and over: 60-...





## Câu 7:

**Tổng hợp số lượng tai nạn theo Mục Đích Hành Trình (Journey Purpose) và Loại Phương Tiện (Vehicle\_Type).**



A picture containing bar chart

Description automatically generated

## Câu 8:

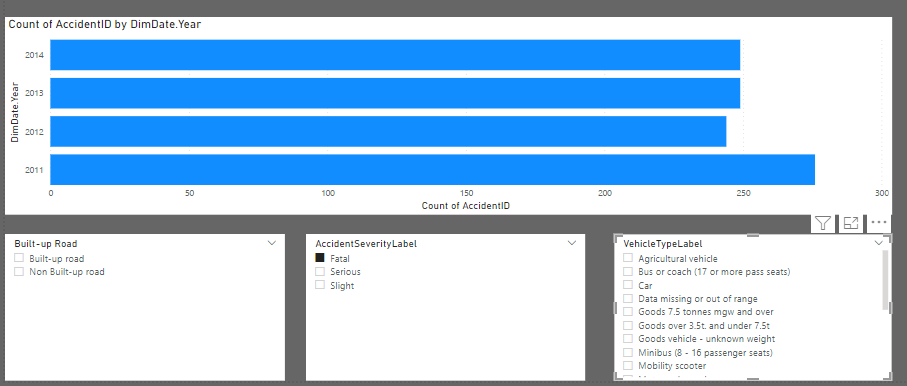
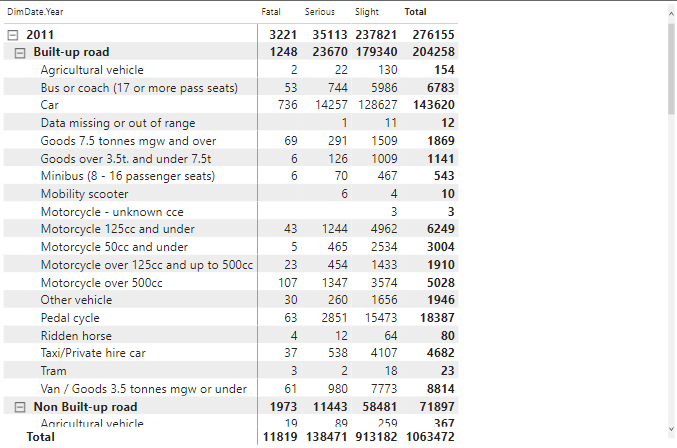
**Tạo thêm thuộc tính Built-up Road trong table Accidients. Built-up Road có 2 giá trị:**

**Built-up road: Nếu tốc độ giới hạn (Speed Limit) dưới 50 mph**

**Non Built-up road: Nếu tốc độ giới hạn từ 50 mph**

## Câu 9:

**Thống kê số lượng tai nạn theo Mức Độ Nghiêm Trọng, Loại Phương Tiện (Vehicle Type), Built-up Road trong các năm.**



## Câu 11:

Tạo 2 Measure trong bảng FactVarience, Accident vs Accident LY

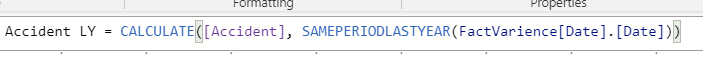
Graphical user interface, application, Word

Description automatically generated

Text

Description automatically generated with medium confidence

Accident: Tính tổng số lượng tai nạn theo từng năm



Accident LY: Lấy số lượng tai nạn của năm trước đó để so sánh/Xem tỉ lệ tăng/giảm



Ratio: % tai nạn tăng/giảm theo các năm

Chart

Description automatically generated

Có thể thấy, từ năm 2011 tới năm 2013 số lượng tai nạn có giảm đi, tuy nhiên qua năm 2014 có dấu hiệu tăng lại, chúng ta nên tìm hiểu lí do tại sao (có thể cơ sở hạ tầng, thiết bị hỗ trợ xuống cấp,…)

Số lượng tai nạn xảy ra chủ yếu ở England

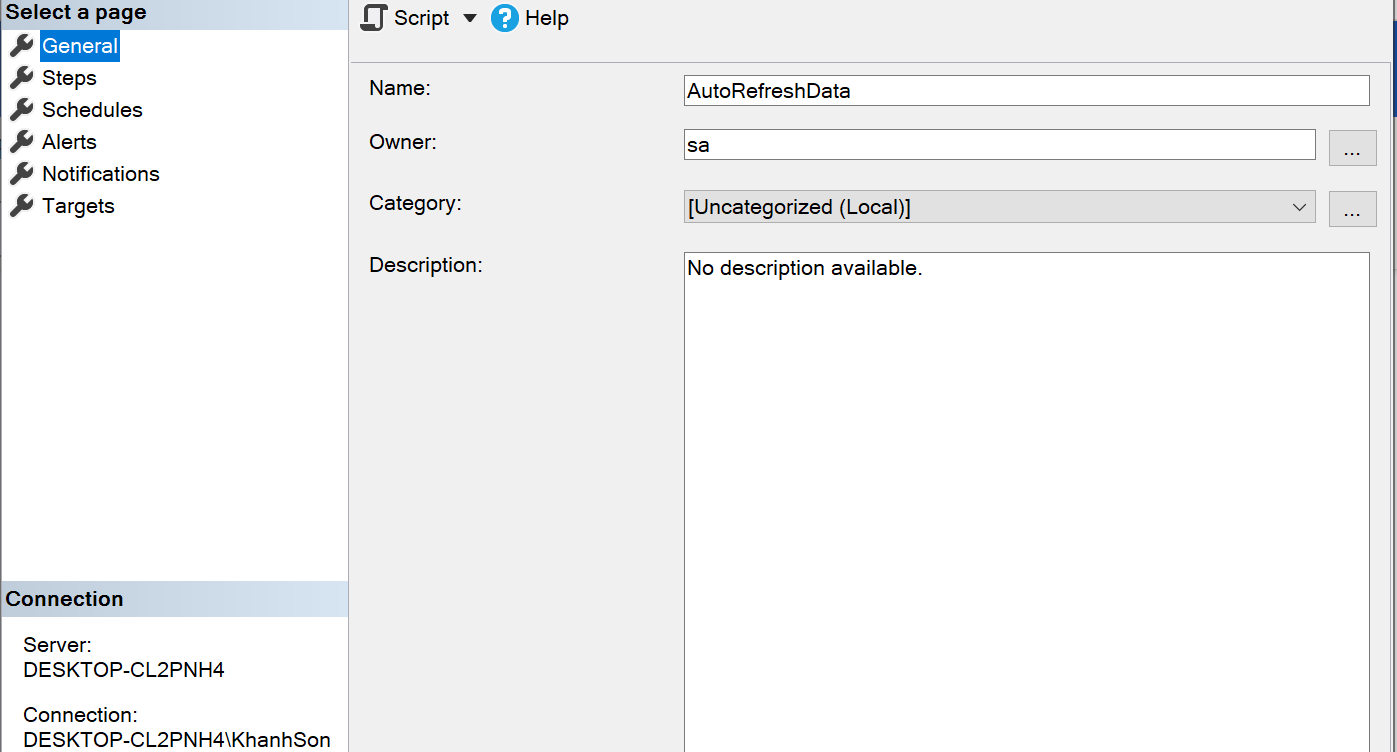
# **LẬP LỊCH THAO TÁC TỰ ĐỘNG ETL**

**Bước 1:** Khởi chạy SQL Server Agent ở chế độ automatic



**Bước 2:** Vào SQL Server Management Studio, login vào Database Engine. Tạo một Job mới trong SQL Server Agent:

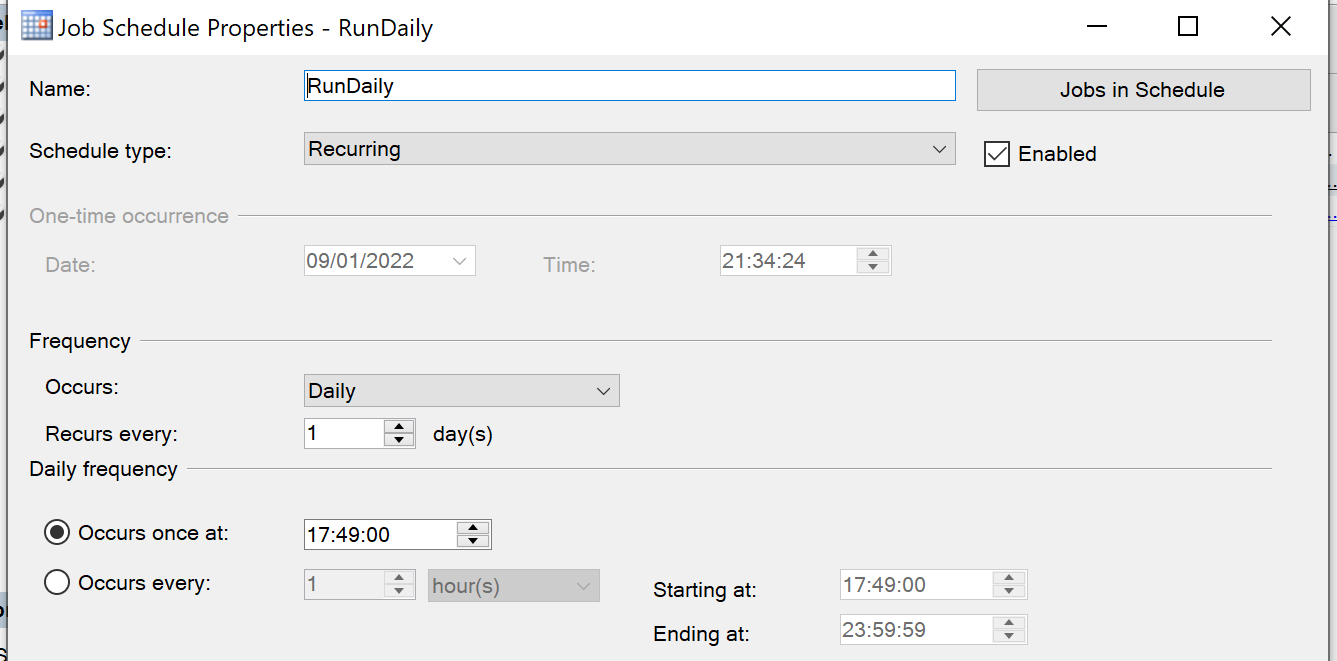
**Bước 3:** Cấu hình lịch



Tại mục general đặt tên cho lịch,và chọn Owner ở đây ta đặt tên lịch là AutoRefreshData và owner là SA.



Tại mục step đặt tên cho các bước cần khởi tạo



Tại mục Schedule ta tiến hành chọn cấu hình cho lịch, cụ thể ở Schedule type chọn Recurring để lịch chạy tự động theo đệ quy, tại phần Frequency ta có thể chọn daily, weekly hay monthly tuỳ theo nhu cầu rồi chọn OK để tiến hành hoàn tất cấu hình lịch

**Bước 4**: Khởi chạy

Ta chọn vào job -> Chọn start job để khởi chạy lịch.