**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**



**Đề tài**

**PROJECT**

**UK ACCIEDENT**

**NHÓM 06**

**PHẠM XUÂN TÚ – 18127056**

**NGUYỄN TIẾN AN - 18127057**

**PHAN NGỌC THIÊN ÂN – 18127037**

**TRƯƠNG PHÚC KHANG – 18127117**

Môn: HTTT Phục Vụ Trí Tuệ Kinh Doanh

Thành phố Hồ Chí Minh – 2022

# MỤC LỤC

[1) MỤC LỤC 2](#_Toc92309781)

[2) CHI TIẾT PHÂN CÔNG 3](#_Toc92309782)

[3) DATA DESCRIPTION 5](#_Toc92309783)

[4) ETL TO STAGE 9](#_Toc92309784)

[4.1 Cách 1 : Sử dụng Truncate thay vì Incremental Extract: 9](#_Toc92309785)

[4.2 Cách 2 : Sử dụng incremental extract bằng việc lấy thời gian LSET và CET: 16](#_Toc92309786)

[5) DATA DESCRIPTION (ETL TO NDS) 26](#_Toc92309787)

[6) ETL STAGE TO NDS 37](#_Toc92309788)

[6.1 Hình ảnh toàn bộ package chuyển từ Stage sang NDS: 37](#_Toc92309789)

[6.2 Ví dụ bước chuyển Accidents từ stage sang NDS: 37](#_Toc92309790)

[6.3 Ví dụ bước chuyển Casualties từ Stage sang NDS: 55](#_Toc92309791)

[7) THIẾT KẾ DDS 71](#_Toc92309792)

[7.1 Miêu tả bảng Fact và các Dimension: 71](#_Toc92309793)

[7.2 Các bảng Dimension sẽ gồm có: 71](#_Toc92309794)

[8) ETL NDS TO DDS 75](#_Toc92309795)

[8.1 Hình ảnh toàn bộ package: 75](#_Toc92309796)

[8.2 Các bước trong ETL NDS-> DDS: 76](#_Toc92309797)

[9) VISUALIZE DATA BẰNG EXCEL PIVOT TABLE VỚI BIỂU ĐỒ VÀ MDX QUERY 94](#_Toc92309798)

[10) DATA MINING 106](#_Toc92309799)

[11) TÀI LIỆU THAM KHẢO 110](#_Toc92309800)

# CHI TIẾT PHÂN CÔNG

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Họ và tên** | **Công việc** | **% hoàn thành** |
| **18127037** | Phan Ngọc Thiên Ân | * ETL Stage --> NDS * ETL NDS --> DDS * Calculated measure | 100 |
| **18127117** | Trương Phúc Khang | * Thiết kế, tạo csdl Stage, NDS sqlserver * Mô tả dữ liệu NDS, DDS * Viết truy vấn MDX * Viết báo cáo | 100 |
| **18127056** | Phạm Xuân Tú | * Thiết kế DDS, tạo csdl DDS sqlserver * Olap ( Phân cấp chiều, build olap cube) | 100 |
| **18127057** | Nguyễn Tiến An | * Phân tích, ETL source --> Stage * Visualize data sử dụng các report tool. * Data mining (Decision Tree) | 100 |

# DATA DESCRIPTION

##### **Phân tích các thuộc tính cần thiết:**

##### Khóa chính của các bảng để có thể thực hiện việc liên kết các bảng.

##### Các thuộc tính cần thiết khác theo từng bảng để thực hiện yêu cầu phân tích:

**Casualty :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| Accident\_index | Char | 0-13 | Mã số vụ tai nạn |
| Casual\_Severity | int | 1-3 | Mức độ thương vong (1: chết, 2: nghiêm trọng, 3: nhẹ) |
| Casualty\_Gender | String | (-1)-2 | Giới tính (-1: lỗi data, 1: Nam, 2: Nữ ) |
| Casualty\_Type | int | 0-98 | Loại tai nạn (0: Pedestrian, 1: Cyclist, 2: Motorcycle 50cc and under rider or passenger, ...) |
| Age\_Band\_of\_Casualty | int | -1 - 11 | Nhóm tuổi thương vong (-1: Null, 1: 0-5, 2: 6-10, … 11: Over 75 ) |
| Age Group | Char | Children - 60 and over | Phân loại độ tuổi gặp tai nạn (Children: 0-15 tuổi, … 60 and over: 60+) |

**Accidents:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| Accident\_index | Char | 0-13 | Mã số vụ tai nạn |
| Local\_Authority (District) | int | 1-941 | Chính quyền địa phương (1: westminster, 2: Camden, 3: Islington, ...) ,theo quận |
| Local\_Authority (Highway) | String | S12000033-E06000014 | Chính quyền địa phương (S12000033: Aberdeen City, S12000034: Aberdeenshire, S12000041: Angus, ...) ,theo cao tốc |
| Date | Date | 1/1/2011-31/12/2014 | Ngày xảy ra sự việc |
| Accident\_Severity | int | 1-3 | Mức độ nghiêm trọng của tai nạn (1: chết, 2: nghiêm trọng, 3: nhẹ) |
| Time | String | 0:00-23:59 | Thời gian xảy ra tai nạn |
| Urban\_or\_Rural Area | int | 1-3 | Địa điểm tai nạn (1: Thành thị, 2: nông thôn, 3: không xác định) |
| Road\_type | int | (-1)-12 | Loại đường gặp tai nạn (1: Roundabout, 2:One way street, 3:Dual carriageway, ... ) |
| Location\_Easting\_OSGR | Int | 166000- 834000 | Độ kinh đông |
| Location\_Northing\_OSGR | Int | 166000 – 834000 | Độ vĩ bắc |
| Longitude | Int | -180 – 180 | Kinh độ |
| Latitude | Int | -180 – 180 | Vĩ độ |
| LSOA | Char |  | Dùng để xác định vị trí của tai nạn |
| Speed\_Limit | Int | 10-70 | Giới hạn tốc độ |

**Vehicles:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| Accident\_index | Char | 0-13 | Mã số vụ tai nạn |
| Journey\_purpose | int | (-1) - 15 | Mục đích chuyến đi (1: Journey as part of work, 2: Commuting to/from work, 3: Taking pupil to/from school, ...) |
| Vehicle\_Type | int | (-1) - 98 | Loại phương tiện (1: Pedal cycle, 2: Motorcycle 50cc and under, 3: Motorcycle 125cc and under, ...) |

##### **Postcode:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| PostCode | Char |  | Mã bưu điện |
| Easting | Int | 166000- 834000 | Độ kinh đông |
| Norting | Int | 166000 – 834000 | Độ vĩ bắc |
| Longitude | Int | -180 – 180 | Kinh độ |
| Latitude | Int | -180 – 180 | Vĩ độ |
| City | Char |  | Tên thành phố |
| County | Char |  | Quận/hạt |
| Country | Char |  | Quê nhà |
| Region | Char |  | Khu vực |

**LSOA**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| PostCode | Char |  | Mã bưu điện |
| LSOA | Char |  | Vị trí của tai nạn |

## 

|  |
| --- |
| ETL TO STAGE |

## Cách 1 : Sử dụng Truncate thay vì Incremental Extract:

##### ***B1*** : Chọn control flow => Create execute sql task => chọn tên cho execute task Sau khi tạo Truncate cho 4 bảng : Accidents, Vehicles, Casualties, Postcode, LSOA ta được ảnh dưới đây:

##### 

##### Ấn vào execute SQL task của Truncate Accidents để có thể chỉnh sửa các thông tin cơ bản như :

##### Name : Tên execute SQL task

##### ConnectionType : OLE DB (thường là mặc định)

##### Connection : Tên của server của máy

##### SQLstatement : Câu lệnh để thực thi execute

##### 

##### ***B2***: Chuyển data từ file CSV vào Database Stage.

##### 

##### Tiến hành nhấp vào Flat file Source để tiến hành đọc file CSV. Tạo connection mới bằng cách New.

##### 

##### **Phần column :**

##### External Column : thuộc tính từ file csv nguồn.

##### Output Column : thuộc tính đọc được từ file csv.

##### Trong bảng các thuộc tính, ta có thể click hoặc bỏ click mỗi thuộc tính để thêm vào hoặc không đọc từ file csv.

##### 

##### Trong phần OLE DB detination, ta chọn tên của server mà ta connect tới, sau khi connect đến server thì ta có thể thấy được số bảng đã đọc được từ phần “Name of the table or the view”.

##### 

##### Vào phần connection Manager ta có thể chọn Server ở mục Server name, sau khi chọn Server thì ở mục “Select or enter a database name” ta sẽ đặt tên cho database.

##### 

##### Dưới đây là mục Mappings các thuộc tính với nhau, ta có thể đối chiếu ở bảng dưới để thấy sự rõ ràng hơn.

##### 

##### Đây là câu lệnh ví dụ sau khi ETL thành công:

##### 

##### Khi ta dùng lệnh Truncate table Accidents thì dữ liệu sẽ bị xóa hết trong Stage. Dưới đây là minh họa dữ liệu được kết nối lại.

##### 

##### **Lý do sử dụng Truncate toàn bộ table chứ không dùng incremental extract :**

##### Data chỉ lấy 1 bảng từ 1 nguồn cố định không đổi theo thời gian.

##### Trong file csv không có thuộc tính hỗ trợ thời gian khởi tạo và thời gian cập nhật lần cuối.

##### Nếu muốn sử dụng LSET và CET thì ta cần đưa thông tin file csv vào 1 database tạm trước, điều này sẽ vô tình tăng task thêm việc truncate database tạm rồi thêm file csv vào, trong khi ta có thể truncate database Stage rồi đổ csv vào đó, còn nếu như file csv hoàn toàn mới so với trước thì ta có thể bỏ qua bước truncate để đổ trực tiếp vào Stage.

## ) Cách 2 : Sử dụng incremental extract bằng việc lấy thời gian LSET và CET:

##### Toàn bộ Package :

##### 

##### Các bước thực hiện:

##### ***B1****:* Truncate Sub Database:

##### 

##### ***B2****:* Chuyển data từ Souce (CSV) tới Database phụ (Sub Stage) :

##### 

##### Lấy data từ file source csv :

##### 

##### 

##### Nhận data vào Sub stage, có sử dụng trigger hoặc derived columns để tạo create date và update date:

##### 

##### 

##### ***B3***: Cập nhật thông tin CET (Current extract time) :

##### 

##### ***B4***: Lấy thông tin LSET (Last extract time) và CET chuyển thành biến :

##### 

##### 

##### ***B5***: Chuyển data từ Sub Stage vào Stage :

##### 

##### Đọc thông tin từ sub Stage bằng sql statement để tìm ra những dòng mới bằng LSET và CET:

##### 

##### 

##### Trong bài này, script stage và sub stage là như nhau nên ta không cần set lại column của sub stage. Sử dụng fast load để bỏ qua các trigger cập nhật create date và update date (nếu như dùng trigger để tạo create date và update date) vì những thông tin đó đã được chỉnh sửa trong sub stage.

##### 

##### 

##### ***B6***: Sau khi hoàn thành chuyển data từ sub, cập nhật lại thông tin bảng metadata thuộc tính LSET = CET hiện tại.

##### 

##### Kết quả package (bảng Postcode):

##### 

##### **Một số lưu ý :**

##### Khi sử dụng trigger sẽ làm giảm tốc độ đọc data từ csv vào sub stage RẤT NHIỀU nên ta nên dùng derived columns.

##### Khi đọc data từ csv hoặc sub stage nếu như gặp lỗi không thể convert data thì hãy vào “Advance editor” mục “output column ” để đổi độ dài các thuộc tính lỗi sao cho khớp với source hoặc script.

# DATA DESCRIPTION (ETL TO NDS)

##### **Phân tích các thuộc tính cần thiết:**

##### Khóa chính của các bảng sẽ được dùng thành natural key để có thể thực hiện việc liên kết các bảng.

##### Các thuộc tính cần thiết khác.

* Khóa chính của bảng lúc này sẽ là surrogate key (tên bảng\_ID) với thuộc tính tự tăng.

##### Có thay đổi 1 số thuộc tính trong các bảng và có thêm những bảng mới được trình bày dưới đây :

##### **Casualty:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| Accident\_index | Char | 0-13 | Mã số vụ tai nạn (Natural key) |
| Casualty\_ID | int |  | Thứ tự từng dòng Casualties (Surrogate key) |
| Vehicle\_Reference | int |  | Tham chiếu phương tiện (Natural key) |
| Casualty\_Reference | int |  | Tham chiếu ca thương vong (Natural key) |
| Accident\_ID | int |  | Mã dòng tai nạn |
| Vehicle\_ID | int |  | Mã dòng phương tiện |
| Casual\_Severity | int | 1-3 | Mức độ thương vong (1: chết, 2: nghiêm trọng, 3: nhẹ) |
| Casualty\_Gender | String | (-1)-2 | Giới tính (-1: lỗi data, 1: Nam, 2: Nữ ) |
| Casualty\_Type\_Key | int | 0-98 | Loại tai nạn (0: Pedestrian, 1: Cyclist, 2: Motorcycle 50cc and under rider or passenger, ...) |
| Age\_Band\_Key | int | -1 - 11 | Nhóm tuổi thương vong (-1: Null, 1: 0-5, 2: 6-10, … 11: Over 75 ) |
| Age Group\_Key | int |  | Phân loại độ tuổi gặp tai nạn (Children: 0-15 tuổi, … 60 and over: 60+) |
| PEDESTRIAN\_LOCATION | int |  | Vị trí người đi bộ |
| PEDESTRIAN\_MOVEMENT | int |  | Cách di chuyển người đi bộ |
| PEDESTRIAN\_ROAD\_  MAINTENANCE\_WORKER | int |  | công nhân bảo trì đường dành cho người đi bộ |
| CASUALTY\_CLASS | int |  | Lớp thương vong |
| SOURCE\_ID | int |  | Mã nguồn |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

**Accidents:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| Accident\_index | Char | 0-13 | Mã số vụ tai nạn (Natural key) |
| ACCIDENT\_ID | int |  | Thứ tự từng dòng Accidents (Surogate key) |
| Local\_Authority (District) | int | 1-941 | Chính quyền địa phương (1: westminster, 2: Camden, 3: Islington, ...) ,theo quận |
| Local\_Authority (Highway) | int |  | Chính quyền địa phương ,theo cao tốc |
| DATE\_ACC | Date | 1/1/2011-31/12/2014 | Ngày xảy ra sự việc |
| Accident\_Severity\_Key | int | 1-3 | Mã mức độ nghiêm trọng của tai nạn (1: chết, 2: nghiêm trọng, 3: nhẹ) |
| Time\_Acc | time | 0:00-23:59 | Thời gian xảy ra tai nạn |
| Urban\_or\_Rural Area | int | 1-3 | Địa điểm tai nạn (1: Thành thị, 2: nông thôn, 3: không xác định) |
| Road\_type\_Key | int | (-1)-12 | Mã loại đường gặp tai nạn (1: Roundabout, 2:One way street, 3:Dual carriageway, ... ) |
| Location\_Easting | char | 166000- 834000 | Độ kinh đông |
| Location\_Northing | char | 166000 – 834000 | Độ vĩ bắc |
| Longitude | char | -180 – 180 | Kinh độ |
| Latitude | char | -180 – 180 | Vĩ độ |
| LSOA | Char |  |  |
| Speed\_Limit | Int | 10-70 | Giới hạn tốc độ |
| NUMBER\_OF\_CASUALTIES | int |  | Số thương vong |
| SOURCE\_ID | int |  | Mã nguồn |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

##### **Vehicles:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| Accident\_index | Char | 0-13 | Mã số vụ tai nạn (Natural key) |
| Vehicle\_ID |  |  | Thứ tự từng dòng phương tiện (Surrogate key) |
| Journey\_purpose\_key | int | (-1) - 15 | Mục đích chuyến đi (1: Journey as part of work, 2: Commuting to/from work, 3: Taking pupil to/from school, ...) |
| Vehicle\_Type\_key | int | (-1) - 98 | Loại phương tiện (1: Pedal cycle, 2: Motorcycle 50cc and under, 3: Motorcycle 125cc and under, ...) |
| Vehicle\_Reference | int |  | Tham chiếu phương tiện (Natural key) |
| SOURCE\_ID | int |  | Mã nguồn |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

##### **Postcode:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| PostCode | Char |  | Mã bưu điện (Natural key) |
| POSTCODE\_ID | int |  | Thứ tự từng dòng postcode (Surrogate key) |
| Easting | Char | 166000- 834000 | Độ kinh đông |
| Norting | Char | 166000 – 834000 | Độ vĩ bắc |
| Longitude | Char | -180 – 180 | Kinh độ |
| Latitude | Char | -180 – 180 | Vĩ độ |
| City | Char |  | Tên thành phố |
| County | Char |  | Quận/hạt |
| Country\_Key | int |  | Mã quê nhà |
| Region | Char |  | Khu vực |
| ISO\_3166\_2 | Char |  |  |
| SOURCE\_ID | int |  | Mã nguồn |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

##### 

##### **LSOA\_INFO:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| POSTCODE\_S | Char |  | Mã bưu điện (Natural key) |
| POSTCODE\_7 | Char |  | Mã bưu điện (Natural key) |
| POSTCODE\_8 | Char |  | Mã bưu điện (Natural key) |
| LSOA\_INFO\_ID | INT |  | Thứ tự từng dòng LSOA INFO (Surrogate key) |
| LSOA | Char |  | Độ kinh đông |
| SOURCE\_ID | int |  | Mã nguồn |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

##### 

**CASUAL\_TYPE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| CASUAL\_TYPE\_ID | int |  | Thứ tự từng dòng Casual\_type (Surrogate key) |
| CASUAL\_TYPE\_KEY | int |  | Mã thương vong (Natural key) |
| CASUAL\_TYPE | Char |  | Thương vong |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

##### 

**AGE\_BAND**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| AGE\_BAND\_ID | int |  | Thứ tự từng dòng Age\_band (Surrogate key) |
| AGE\_BAND\_KEY | int |  | Mã độ tuổi (Natural key) |
| AGE\_BAND | Char |  | Độ tuổi |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

**AGE\_GROUP**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| AGE\_GROUP\_ID | int |  | Thứ tự từng dòng Age\_group (Surrogate key) |
| AGE\_GROUP\_KEY | int |  | Mã nhóm tuổi (Natural key) |
| AGE\_GROUP | Char |  | Nhóm tuổi |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

##### 

**CASUAL\_SEVERITY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| CASUAL\_SEVERITY\_ID | int |  | Thứ tự từng dòng CASUAL\_SEVERITY (Surrogate key) |
| CASUAL\_SEVERITY\_KEY | int |  | Mã thương vong (Natural key) |
| CASUAL\_SEVERITY | Char |  | Thương vong |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

**VEHICLES\_TYPE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| VEHICLES\_TYPE\_ID | int |  | Thứ tự từng dòng VEHICLES\_TYPE (Surrogate key) |
| VEHICLES\_TYPE\_KEY | int |  | Mã loại phương tiện (Natural key) |
| VEHICLES\_TYPE | Char |  | Phương tiện |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

**JOUNEY\_PURPOSE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| JOUNEY\_PURPOSE\_ID | int |  | Thứ tự từng dòng JOUNEY\_PURPOSE (Surrogate key) |
| JOUNEY\_PURPOSE\_KEY | int |  | Mã mục đích chuyến đi (Natural key) |
| JOUNEY\_PURPOSE | Char |  | mục đích chuyến đi |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

**ROAD\_TYPE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| ROAD\_TYPE\_ID | int |  | Thứ tự từng dòng ROAD\_TYPE (Surrogate key) |
| ROAD\_TYPE\_KEY | int |  | Mã loại đường (Natural key) |
| ROAD\_TYPE | Char |  | loại đường |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

**ACCIDENTS\_SEVERITY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| ACCIDENTS\_SEVERITY\_ID | int |  | Thứ tự từng dòng ACCIDENTS\_SEVERITY (Surrogate key) |
| ACCIDENTS\_SEVERITY\_KEY | int |  | Mã mức độ tai nạn (Natural key) |
| ACCIDENTS\_SEVERITY | Char |  | mức độ tai nạn |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

**COUNTRY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| COUNTRY\_ID | int |  | Thứ tự từng dòng COUNTRY (Surrogate key) |
| COUNTRY\_KEY | int |  | Mã nước (Natural key) |
| COUNTRY | Char |  | Tên nước |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

##### 

##### **SOURCE:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Miền giá trị** | **Ý nghĩa** |
| SOURCE\_ID | int |  | Thứ tự từng dòng SOURCE (Surrogate key) |
| SOURCE\_KEY | int |  | Mã nguồn (Natural key) |
| SOURCE | Char |  | Tên nguồn |
| Create\_Date | Date |  | Ngày tạo |
| Update\_Date | Date |  | Ngày cập nhật |

# ETL STAGE TO NDS

## Hình ảnh toàn bộ package chuyển từ Stage sang NDS:

##### 

## **Ví dụ bước chuyển Accidents từ stage sang NDS:**

##### 

##### 

##### **Load Accidents Stage :**

##### Load lên tất cả các cột thông tin từ database Stage bảng Accidents bình thường

##### 

##### **Add Source**

##### Thêm vào trong dữ liệu 1 thuộc tính Source ID để lưu trữ thông tin nguồn dữ liệu, tại đâu ta để 1 vì nhóm em quy định source flat file có mã là 1.

##### 

##### **Look up Accident Severity**

##### Tìm xem thuộc tính Accident Severity trong dòng ta đang muốn add và NDS đã tồn tại trong bảng Accident Severity mà đã được tạo (add) vào từ trước hay chưa. Tại đây vì nhóm không có add thông tin Accident Severity từ 1 flat file nguồn (giống bảng Accidents, Vehicles, Casualties,...) mà tự thêm vào bằng việc xem xét UK Accident codebook nên bảng Accident Severity không có Source ID. Sau khi tìm ra giá trị khớp với thông tin từ bảng Accident Severity (Accident Severity của Accidents trong Stage tham chiếu tới Accident\_Severity\_key của Accident Severity trong NDS) thì ta sẽ trả lại Accident\_Severity\_ID của thuộc tính đó (khóa chính của bảng Accident Severity) để có thể đưa vào bảng Accidents trong NDS nhằm tham chiếu khóa ngoại 2 bảng trong NDS với nhau.

##### 

##### 

##### Ta sẽ tìm các giá trị match trong bảng Accident Severity còn nếu có xuất hiện giá trị không match thì đó sẽ là lỗi sẽ xử lý sau.

##### Tương tự với Look up Road Type và Look up Local Highway, ta cũng tìm thuộc tính có trong 2 bảng có sẵn đó trong NDs rồi trả về giá trị khóa chính của 2 bảng đó để tham chiếu trong Accident trong NDS.

##### 

##### 

##### 

##### 

##### **Look up LSOA:**

##### Look up Lsoa thì có chút đặc biệt hơn vì ta không tìm trong Lsoa\_info để lấy lại Lsoa\_info\_ID để tham chiếu trong NDS bởi ở trong Stage thì Lsoa\_info và Accidents không thể tham chiếu thuộc tính Lsoa cho nhau vì 2 thuộc tính đó đều không phải duy nhất ở trong 2 bảng nên không thể tham chiếu theo khóa ngoại được. Mà ta tìm Lsoa của Accidents trong Lsoa\_info vì ta cần xác định được thông tin Lsoa của tai nạn này có nằm trong bảng Lsoa\_info hay không để thuận tiện hơn cho việc liên kết với postcode, nhằm tìm ra vị trí chính xác của địa điểm xảy ra tai nạn tính đến county, city, …

##### 

##### 

##### **Look up existing Accidents:**

##### Tại đây ta sẽ tìm xem dòng dữ liệu ta muốn đưa vào NDS đã có sẵn trong đó hay chưa bằng việc tìm so sánh khóa chính của bảng Accidents trong Stage với khóa tự nhiên của Accients trong NDS. Nếu tìm thấy thì có nghĩa đã có dòng này trong NDS, ta cần update mới lại dòng đó; còn nếu không tìm thấy nghĩa là đây là 1 dòng mới cần insert vào trong NDS. Và vì ta đang nhập Accients vào từ source là flat file nên ta chỉ tìm các dòng trùng trong cùng source = 1 đó.

##### 

##### 

##### **Update lại dòng cũ :**

##### Tạo 1 ngày update mới bằng ngày hôm nay:

##### 

##### Cập nhật lại dòng đó trong NDS với câu lệnh UPDATE:

##### 

##### 

##### **Lưu ý** : các giá trị nào đã được lookup ở trên ngoài Lsoa và Accidents thì cần thay thế bằng ID khóa chính tìm ra của bảng đó.

##### 

##### **Insert 1 dòng mới :**

##### Tạo ngày created bằng ngày hôm nay:

##### 

##### Insert vào bảng Accidents trong NDS:

##### 

##### **Lưu ý:** các giá trị nào đã được lookup ở trên ngoài Lsoa và Accidents thì cần thay thế bằng ID khóa chính tìm ra của bảng đó.

##### . .

## Ví dụ bước chuyển Casualties từ Stage sang NDS:

##### 

##### **Load Casualties từ Stage:**

##### 

##### 

##### **Add Source** (Flat file = 1):

##### 

##### **Look up Age Group** (So sánh Age\_Group\_Key và Lấy Age Group ID):

##### 

##### 

##### **Look up Age Band** (So sánh Age\_Band\_Key và Lấy Age Band ID):

##### 

##### 

##### **Look up Casualty Type** (So sánh Casual Type Key, lấy Casual Type ID):

##### 

##### 

##### **Look up Casualty Severity** (So sánh Casual Severity Key,lấy Casual Severity ID):

##### 

##### 

##### **Look up Accident** (Source\_ID=1, so sánh Accident Index, Lấy ra Accident\_ID):

##### 

##### 

##### **Look up Vehicle** (Source\_ID=1, so sánh Accident Index và Vehicle Reference, Lấy ra Vehicle\_ID):

##### 

##### 

##### **Look up existing Casualties** (Source\_ID=1, so sánh Accident Index, Vehicle Reference và Casualty Reference) nếu dòng đã có tồn tại thì ta update, chưa có thì ta insert vào NDS:

##### 

##### 

##### **Update lại dòng cũ:**

##### Cập nhật lại ngày UPDATE:

##### 

* + Update lại dòng bằng lệnh UPDATE:

##### 

##### 

##### **Insert 1 dòng mới:**

##### Cập nhật lại ngày Created\_Date:

##### 

##### Insert vào bảng Casualties trong NDS:

##### 

##### 

##### **Lưu ý**: Khi sử dụng lookup nhớ chỉnh khi không tìm ra giá trị match thì sẽ redirect row sang error hoặc sang no match chứ không để fail component vì điều đó sẽ làm package dừng lại khi gặp 1 giá trị không hợp lệ như NULL,...

##### 

##### Các bước load bảng khác thì làm tương tự như 2 ví dụ trên, không có gì khác biệt.

# THIẾT KẾ DDS

## Miêu tả bảng Fact và các Dimension:

##### Sẽ có 3 bảng Fact gồm có Fact\_Casualty lấy bảng Casualities trong NDS làm fact, bảng Fact\_Accident lấy bảng Accidents trong NDS là fact và bảng Fact\_Vehicle lấy bảng Vehicles trong NDS làm fact.

## Các bảng Dimension sẽ gồm có:

##### **Fact Casualty:**

##### Dim Accident.

##### Dim Date.

##### Dim Local\_Authority\_District.

##### Dim Age\_Band.

##### Dim Age\_Group.

##### Dim Gender.

##### Dim Casual\_Severity.

##### Dim Casual\_Type.

##### 

##### **Fact Accident:**

##### Dim Date.

##### Dim Time\_Period.

##### Dim Accident\_Severity.

##### Dim Road\_Type.

##### Dim Urban\_Rural.

##### 

##### **Fact Vehicles:**

##### Dim Accident.

##### Dim Accident\_Severity.

##### Dim Date.

##### Dim Build\_Up\_Road.

##### Dim Journey\_Purpose.

##### Dim Vehicle\_Type.

##### 

# ETL NDS TO DDS

## Hình ảnh toàn bộ package:

##### 

##### 

## Các bước trong ETL NDS-> DDS:

##### **Bước 1:**

##### Get Cet và Lset của các bảng muốn ETL vào trong DDS với giá trị CET là getdate() và lưu vào biến cục bộ trong SSIS.

##### Lưu ý thêm rằng vì trong NDS đa số sẽ là các bảng được tạo bằng tay theo Codebook (accident severity, vehicle type,...) chứ không phải được đọc lên từ data nên việc tạo lset và cet chỉ phục vụ cho việc kiểm soát Metadata chứ đôi khi sẽ không cần phải dùng increment etl cho các bảng này.

##### VD: get Cet và lset của dim accidents:

##### 

##### 

##### 

##### 

##### 🡪 Làm tương tự cho các bảng dimension khác.

##### **Bước 2:**

##### **Đọc các bảng Dim vào trong NDS:**

##### Đọc bảng Dim Date:

##### Bảng Dim Date không có sẵn trong NDS cũng như không được cung cấp thông qua bộ dữ liệu trong code book nên ta phải tự tạo bảng Dim Date bằng cách tách thuộc tính từ trường thời gian trong bảng Accidents bằng Derived Columns.

##### 

##### Ta đọc những ngày của các dòng dữ liệu mới trong khoảng từ Lset đến Cet:

##### 

##### Dùng các hàm trong Derived columns để tìm các giá trị ngày, tháng, quý, năm:

##### 

##### Dùng look up kiểm tra xem trong bảng Dim Date đã có ngày mình chuẩn bị add vào hay chưa, nếu không có thì mới add vào còn nếu đã tồn tại thì bỏ qua dòng đó.

##### 

##### Đọc vào trong bảng Dim Date với mapping tương ứng, nên có 1 thuộc tính có kiểu Datetime để giữ lại ngày đúng được đọc ra từ bảng Accidents để có thể tiện cho việc nhìn nhận data về sau (ở đây chính là thuộc tính Date\_Full\_String).

##### 

* + Đọc bảng Dim Accidents:

##### Bảng Dim Accident sẽ được đọc theo kiểu Increment vì bảng này được đọc từ data source chứ không phải tự tạo. Increment Accident:

##### Ta cần merge giá trị bảng Local Authority District và Postcode vào để có thể dùng cho việc thống kê vị trí nhằm vẽ reginal map về sau, ngoài ra thì 3 bảng Time Period, Accident Severity Local Authority District còn là dùng để tham chiếu giá trị cho 2 bảng fact Vehicle và Casualty.

##### 

##### 

##### Cách Merge Join bảng Accidents với các bảng khác:

##### Để Merge Join thì đầu tiên ta phải Sort lại các giá trị trong bảng theo thuộc tính mà ta dùng để Merge.

##### Trong trường hợp Merge với Local Authority District thì thuộc tính là Local Authority District ID (Accident) và Local Authority District ID (Local Authority District).

##### Trong trường hợp Merge với Time Period thì thuộc tính đó là Time Period ID (Accident) và Time Period ID (Time Period).

##### Trong trường hợp Merge với Accident Severity thì thuộc tính là Accident Severity ID (Accident) và Accidnet Severity ID (Accident Severity).

##### Trong trường hợp Merge với Postcode thì thuộc tính đó là Local Authority District (Accident) và City (Postcode) .

##### Khi Merge join các bảng thì ta sẽ lấy được các giá trị của bảng ngoài Accident luôn chứ không chỉ là ID.

##### Vd: Sort và merge join giữa Accident với Local Authority District

##### 

##### 

##### Chỉnh chế độ Merge Join là Left Outer Join để giữ lại bảng Accidents và thêm thông tin bảng Local Authority District vào bảng Accidents.

##### 

##### Đọc các bảng Dim còn lại:

##### Các bảng Dim còn lại vì được tạo trong NDS nên ta sẽ đọc thẳng một cách bình thường mà không cần xét Lset và Cet.

##### VD: ETL bảng Accident Severity từ NDS và Dim Accident Severity trong DDS

##### 

##### 

##### 

##### 

* Vì trong NDS ta đã look up để tạo các khóa ngoại với surrogate key nên trong DDS ta không cần tạo lại Surrogate key bằng Identity(1,1) nữa mà chỉ cần đọc thẳng ID vào.

##### **Bước 3:**

##### **Đọc bảng Fact vào trong DDS**

##### Đọc bảng Casualties vào làm bảng Fact\_Casualty:

##### 

##### Ta cần look up thông tin của Local Authority District, Date vì trong bảng Casualties không có thuộc tính đó mà các thuộc tính kia tồn tại trong bảng Accident bằng Accident Index sẵn có. Ngoài ra ta cần look up lại Accidents ID mặc dù trong bảng Casualties đã có vì khi trong dds thì Dim Accidents được đọc theo kiểu Increment nên có tỉ lệ các bảng mới được đọc vào và thay đổi giá trị ID nên cần look up lại để tránh sai sót.

##### 

##### Tiếp theo ta sẽ look up ngày vừa tìm được từ bảng Accident để tìm ra DateID trong bảng Dim Date:

##### 

##### Và cuối cùng ta sẽ kiểm tra xem dòng dữ liệu này đã có trong bảng DDS hay chưa, nếu đã có rồi thì ta sẽ update lại các thông tin còn nếu chưa có ta sẽ insert mới vào.

##### 

##### Đọc bảng Accidents vào làm bảng Fact\_Accident:

##### 

##### Khác với bảng Fact Casualty thì các báo cáo yêu cầu bảng Fact Accident lại không cần thêm các thuộc tính nằm ngoài bảng nào nên ta không cần look up các thuộc tính ở ngoài bảng mầ chỉ cần look up date id từ Dim Date mà thôi.

##### 

##### Và cuối cùng ta sẽ kiểm tra xem dòng dữ liệu này đã có trong bảng DDS hay chưa, nếu đã có rồi thì ta sẽ update lại các thông tin còn nếu chưa có ta sẽ insert mới vào.

##### 

##### Đọc bảng Vehicles vào làm bảng Fact\_Vehicle:

##### 

##### Tương tự như trong Fact Casualty, bảng Fact Vehicle khi thống kê báo cáo thì lại cần một số thuộc tính không có sẵn trong bảng Vehicle mà lại có trong bảng Accident nên ta cần look up trong bảng Accident để tìm các ID thuộc tính đó (ở đây là thuộc tính DateAcc giữ ngày tai nạn, Build Up Road giữ vận tốc và Accident Severity key giữ độ nghiêm trọng của tai nạn).

##### **Lưu ý** : Ta vẫn cần tìm lại Accident ID để phòng khi khi đọc vào DDS thì increment ETL hoặc sau khi sort làm ID của bảng Accident bị đổi.

##### 

##### Sau đó ta dùng DateAcc vừa tìm được để tìm ra Date ID trong bảng Dim Date:

##### 

##### Và cuối cùng ta sẽ kiểm tra xem dòng dữ liệu này đã có trong bảng DDS hay chưa, nếu đã có rồi thì ta sẽ update lại các thông tin còn nếu chưa có ta sẽ insert mới vào.

##### 

# VISUALIZE DATA BẰNG EXCEL PIVOT TABLE VỚI BIỂU ĐỒ VÀ MDX QUERY

##### 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:

##### MDX query và kết quả câu truy vấn:

##### 

##### 

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

##### MDX query và kết quả câu truy vấn:

##### 

##### 

##### 3) 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:

##### Ảnh có chứa văn bản Mô tả được tạo tự động

##### MDX query và kết quả câu truy vấn:

##### 

##### 4) 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.

##### MDX query và kết quả câu truy vấn:

##### 

##### 

##### 5) Thống kê số lượng TNGT theo Mức Độ Nghiêm Trọng, Vùng thành thị hay nông thôn (Urban\_or\_Rural\_Area), và Kiểu Đường (Road Type) trong các năm.

##### MDX query và kết quả câu truy vấn:

##### 

##### 

##### 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

##### 60 and over: 60-...

##### 

##### 

##### MDX query và kết quả câu truy vấn:

##### 

##### 

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

##### MDX query và kết quả câu truy vấn:

##### 

##### 

##### 9) Thống kê số lượng tai nạn theo Mức Độ Nghiêm Trọng, Loại Phương Tiện

##### (Vehicle Type), Built-up Road trong các năm.

##### MDX query và kết quả câu truy vấn:

##### 

##### 10) Thống kê số lượng phương tiện trong tai nạn theo độ nghiêm trọng của tai nạn trong các năm.

##### Ảnh có chứa văn bản, bàn Mô tả được tạo tự động

##### 

##### MDX query và kết quả câu truy vấn:

##### 

##### 

##### 10) Định nghĩa fact Variance để tính mức độ tăng giảm của TNGT theo đơn vị phần trăm qua các năm:

##### Ảnh có chứa văn bản, bàn Mô tả được tạo tự động

##### 

##### 13) Dùng regional map để biểu diễn trực quan (bằng màu sắc) số lượng TNGT ở các vùng trong năm (một số county không thể thống kê vì tên tỉnh được lưu khác nhau trong bảng Postcode và thuộc tính local authority district trong bảng Accidents (ví dụ như trong local authority district có tên Carmen nhưng trong Postcode thì tỉnh đó lại có tên là Carmen town)).

##### Ảnh có chứa bàn Mô tả được tạo tự động

##### **Ghi chú :** Một vài câu không thể thể chart vì dữ liệu quá nhiều(số lượng dòng và cột), dẫn đến khi xuất ra thì chúng ta rất khó nhìn và thống kê.

# DATA MINING

##### **Trường hợp :** Dự đoán độ nghiêm trọng (Fatal, Serious, Slight) của tai nạn.

##### Mô hình sử dụng : Decision Tree

##### Bảng sử dụng : Fact-Accident

##### Thuộc tính sử dụng : UrbanRuralID, RoadTypeID và TimePeriodID

##### 1. Mining structure :

##### - Mining model : Decision tree 🡪Từ bảng Fact Accident ta set :

##### - Accident Index : là khoá.

##### - Road Type ID, Time Period ID, UrbanRural ID : là thuộc tính input.

##### - Accident Severity ID : thuộc tính được dự đoán.

##### 

##### 2. Mining model viewer : - Đây là cây quyết định được tự động generate ra gồm 2 thuộc tính là Road Type và Time Period.

##### 3. Mining model prediction :

##### Ảnh có chứa văn bản, ảnh chụp màn hình, trong nhà Mô tả được tạo tự động

##### 

##### 🡪Từ Accident ID thì kết quả cho ra dự đoán về mức độ nghiêm trọng của tai nạn.

##### 4. Mining legend :

##### 

##### Coefficient: Giá trị được nhân với giá trị dự đoán (Prediction values).

##### Histogram: 1 phương pháp đồ hoạ để hiển thị dữ liệu định lượng.

# TÀI LIỆU THAM KHẢO