ĐỒ ÁN MÔN HỌC HỆ THỐNG THÔNG TIN PHỤC VỤ TRÍ TUỆ KINH DOANH

Author(s): Nhóm 2

*1712674 – Nguyễn Quang Phúc*

*1712679 – Phạm Hồng Phước*

*1712501 – Trần Quang Huy*

*1712594 – Lê Văn Minh*

Table of Contents

[Executive Summary 4](#_Toc60921594)

[Project description 4](#_Toc60921595)

[Key Deliverables 4](#_Toc60921596)

[Critical Success Factors 4](#_Toc60921597)

[Risks and Concerns 4](#_Toc60921598)

[Job Assignment 5](#_Toc60921599)

[Business Requirements 6](#_Toc60921600)

[Requirement 1: 6](#_Toc60921601)

[Requirement 2: 6](#_Toc60921602)

[Requirement 3: 7](#_Toc60921603)

[Requirement 4: 7](#_Toc60921604)

[Data source Description 8](#_Toc60921605)

[Yellow Taxi Trip 8](#_Toc60921606)

[2010 Census Tracts 10](#_Toc60921607)

[Data Warehouse Design 11](#_Toc60921608)

[Stage Design 11](#_Toc60921609)

[NDS Design 12](#_Toc60921610)

[DDS Design 13](#_Toc60921611)

[Data Flow Design 15](#_Toc60921612)

[Data Preparation and Processing 16](#_Toc60921613)

[ETL 19](#_Toc60921614)

[Source to Stage 19](#_Toc60921615)

[Stage to NDS 21](#_Toc60921616)

[**1.** **NDS Cencus Block** 21](#_Toc60921617)

[**2.** **NDS Yellow Taxi Data Trip** 25](#_Toc60921618)

[NDS to DDS 26](#_Toc60921619)

[**1.** **Insert Dim DateTime** 26](#_Toc60921620)

[**2.** **Insert Dim Location** 27](#_Toc60921621)

[**3.** **Insert Fact F\_TaxiTrip** 28](#_Toc60921622)

[OLAP 31](#_Toc60921623)

[Tạo Cube 31](#_Toc60921624)

[Browse trên Cube 33](#_Toc60921625)

[MDX 34](#_Toc60921626)

[KPI SASS 37](#_Toc60921627)

[DATA MINING 38](#_Toc60921628)

[TABLEAU REPORT DASHBOARD 47](#_Toc60921629)

# Executive Summary

## Project description

Tài liệu này dùng để chỉ ra các yêu cầu nghiệp vụ phục vụ cho việc phân tích, báo cáo, dự đoán kinh doanh cho công ty Yellow Taxi Trip.

## Key Deliverables

Các yêu cầu nghiệp vụ cần sử dụng cho việc thực hiện trí tuệ kinh doanh cho công ty.

## Critical Success Factors

Từ các yêu cầu nghiệp vụ có thể thiết kế các bảng Fact, Dim của Kho dữ liệu.

## Risks and Concerns

Các yêu cầu chưa thể bao quát toàn bộ nhu cầu phân tích của công ty.

# Job Assignment

## Giữa kỳ

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Thành viên** | **Công việc** | **Mức độ hoàn thành** |
| 1712674 | Nguyễn Quang Phúc | Mô tả dữ liệu Yellow Taxi Trip  Thực hiện vẽ Data Flow | 100% |
| 1712501 | Trần Quang Huy | Phân tích yêu cầu  Thiết kế Fact Table, Dim Table  Ghi báo cáo | 100% |
| 1712594 | Lê Văn Minh | Mô tả dữ liệu Yellow Taxi Trip  Thiết kế NDS  Thiết kế Fact Table, Dim Table | 100% |
| 1712679 | Phạm Hồng Phước | Mô tả dữ liệu 2010 Cencus Tracts  Thiết kế Fact Table, Dim Table  Thiết kế NDS | 100% |

## Cuối kỳ

|  |  |  |  |
| --- | --- | --- | --- |
| **MSSV** | **Thành viên** | **Công việc** | **Mức độ đóng góp toàn bộ đồ án** |
| 1712674 | Nguyễn Quang Phúc | Đổ dữ liệu từ Source vô Stage  Viết MDX Query trong OLAP | 22% |
| 1712501 | Trần Quang Huy | Đổ dữ liệu từ Source vô Stage  Đổ dữ liệu từ NDS vô DDS  Thực hiên OLAP, KPI in SSAS | 26% |
| 1712594 | Lê Văn Minh | Đổ dữ liệu TaxiTrip từ Stage vô NDS  Đổ dữ liệu từ NDS vô DDS  Thực hiện Data Mining, OLAP | 26% |
| 1712679 | Phạm Hồng Phước | Data Preparation bằng Python  Đổ dữ liệu CencusBlock từ Stage vô NDS  Tạo Report Tableau  Chỉnh format Báo cáo đồ án | 26% |

# Business Requirements

Table 1: List of Business Requirements

|  |  |  |
| --- | --- | --- |
| Identifier | Description | Priority |
| 1 | Thống kê được lượng phân bố đặt xe/xuống xe của từng thời điểm trong ngày ở các khu vực trong các quý/tháng/năm | High |
| 2 | Phân tích số lượng đặt xe/xuống xe của từng khu vực trong các quý/tháng/năm | High |
| 3 | Phân tích doanh thu trong ngày/tháng/quý/năm của các khu vực và toàn bộ công ty | High |
| 4 | Phân tích số lượng các chuyến xe thành công, bị hủy,.. và cho biết tỉ lệ rủi ro… | Medium |

Requirement 1:   
Thống kê được lượng phân bố đặt xe/xuống xe của từng thời điểm (theo giờ) trong ngày ở các khu vực qua các quý, tháng, năm

**Data sources**

* Pickup/Dropoff Datetime (Yellow Taxi Data).
* CTLabel (Census Tracts).

**Business and data transformations**

* Chuyển Pickup/Dropoff Datetime sang Giờ, Ngày, Tháng, Năm.

**Business processes**

* Dùng hàm Count(\*) để tính số lượng đặt xe/xuống xe trong các giờ tại khu vực.

## Requirement 2:

Phân tích số lượng đặt xe/xuống xe của từng khu vực trong các quý/tháng/năm

**Data sources**

* Pickup/Dropoff Datetime (Yellow Taxi Data).
* CTLabel (Census Tracts).

**Business and data transformations**

* Chuyển Pickup/Dropoff Datetime sang Giờ, Ngày, Tháng, Năm.

**Business processes**

* Dùng hàm Count(\*) để tính số lượng đặt xe/xuống xe của các khu vực theo giờ

Requirement 3:Phân tích doanh thu trong ngày/tháng/quý/năm của các khu vực và toàn bộ công ty

**Data sources**

* total\_amount (Yellow Taxi Data).
* Pickup/Dropoff Datetime (Yellow Taxi Data).
* CTLabel (Census Tracts).

**Business and data transformations**

* Chuyển Pickup/Dropoff Datetime sang Giờ, Ngày, Tháng, Năm.

**Business processes**

* Dùng hàm Sum(total\_amount) để tính tổng số tiền của tất cả các chuyến xe được đặt tại một khu vực trong một tháng/năm.

## Requirement 4:

Phân tích số lượng các chuyến xe thành công, bị hủy,.. và cho biết tỉ lệ rủi ro…

**Data sources**

* Payment\_type (Yellow Taxi Data).
* Pickup/Dropoff Datetime (Yellow Taxi Data).

**Business and data transformations**

* Chuyển Pickup/Dropoff Datetime sang Giờ, Ngày, Tháng, Năm
* Success\_Ratio = số chuyến thành công / tổng chuyến
* Fail\_Ratio = số chuyến thất bại / tổng chuyến

**Business processes**

* Đếm những chuyến thành công và thất bại sau đó tính ra tỉ lệ.

# Data source Description

## Yellow Taxi Trip

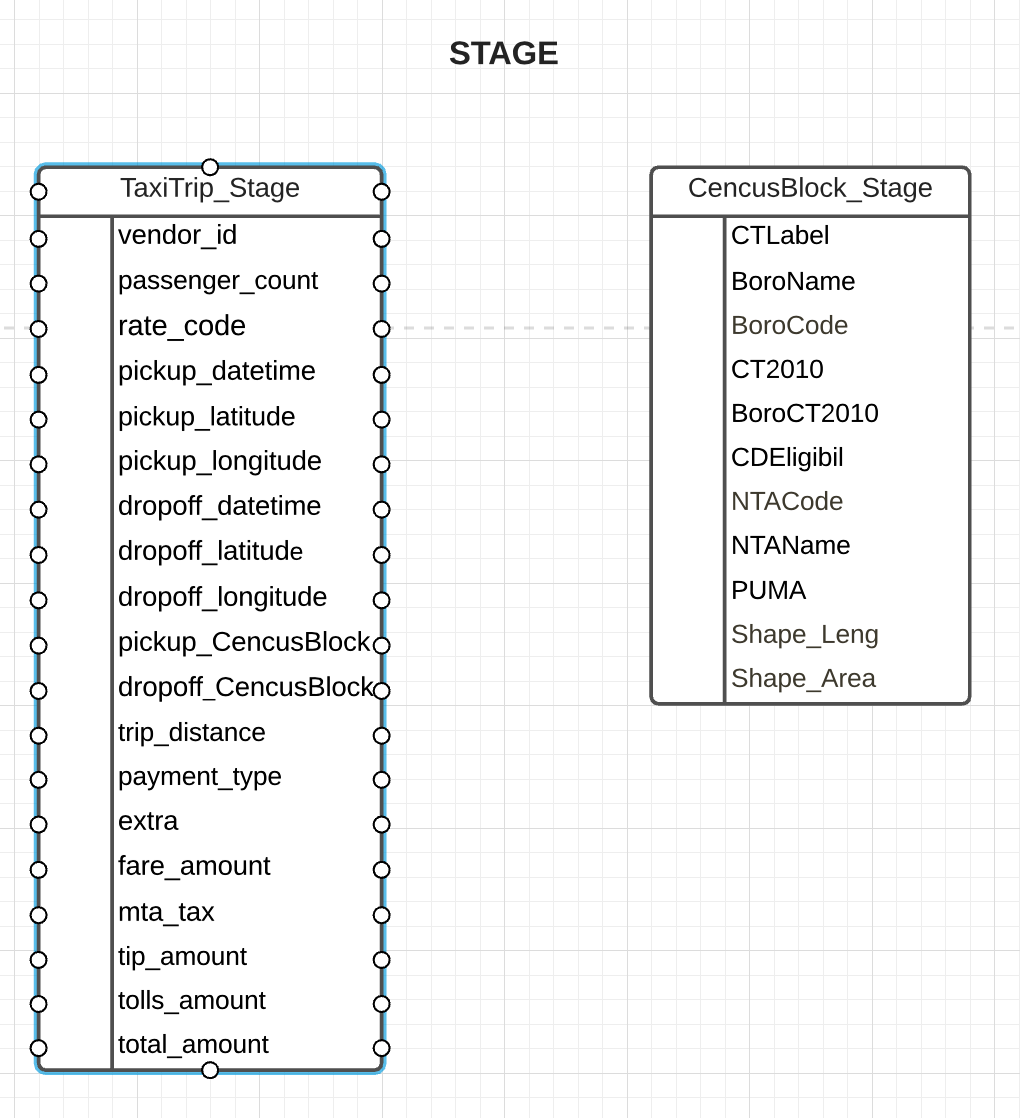
|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Description** |
| dropoff\_datetime | Datetime | Thời điểm đồng hồ công tơ mét ngắt. |
| dropoff\_latitude | Float | Vĩ độ nơi đồng hồ công tơ mét ngắt. |
| dropoff\_longitude | Float | Kinh độ nơi đồng hồ ngắt. |
| extra | Float | Phí phát sinh và phụ phí. Bao gồm 0,5$ và 1$ đối với giờ cao điểm và cuốc xe xuyên đêm. |
| fare\_amount | Float | Số tiền xe taxi tổng cộng đã di chuyển, được tính dựa trên quãng đường và thời gian, được đo bằng đồng hồ công tơ mét ở trên taxi. |
| imp\_surcharge | Float | Một khoản phụ phí, mục đích giúp cải thiện chất lượng dịch vụ tốt hơn. |
| mta\_tax | Float | Thuế MTA. |
| passenger\_count | Int | Số lượng khách trên xe. |
| payment\_type | Int | Mã số biểu thị cách hành khách thanh toán cho chuyến đi. 1 = Thẻ tín dụng, 2 = Tiền mặt, 3 = Không tính phí, 4 = Tranh chấp, 5 = Không xác định, 6 = Chuyến đi bị hủy. |
| pickup\_datetime | Datetime | Thời điểm đón khách. |
| pickup\_latitude | Float | Vĩ độ đón khách. |
| pickup\_longitude | Float | Kinh độ đón khách. |
| rate\_code | Int | Mã giá cuối cùng có hiệu lực vào cuối chuyến đi. 1 = Giá tiêu chuẩn, 2 = JFK, 3 = Newark, 4 = Nassau hoặc Westchester, 5 = Giá thỏa thuận, 6 = Đi theo nhóm. |
| store\_and\_fwd\_flag | Int | Flag cho biết liệu bản ghi chuyến đi có được lưu trong bộ nhớ xe hay không trước khi gửi đến nhà cung cấp |
| tip\_amount | Float | Số tiền tip.  Trường này tự động được điền cho các mẹo về thẻ tín dụng. Tiền boa không được bao gồm. |
| tolls\_amount | Float | Tiền trạm thu phí đường. |
| total\_amount | Float | Tổng số tiền khách hàng phải trả. Không bao gồm tiền tip. |
| trip\_distance | Float | Tổng khoảng cách đi được của chuyển đi (đv mile) |
| vendor\_id | Int | Nhà cung cấp công nghệ đã cung cấp hồ sơ. CMT = Creative Mobile Technologies. VTS = VeriFone, Inc. DDS = Digital Dispatch Systems. |
| RateCodeID | Int | Mã giá cuối cùng có hiệu lực vào cuối chuyến đi. 1 = Giá tiêu chuẩn, 2 = JFK, 3 = Newark, 4 = Nassau hoặc Westchester, 5 = Giá thỏa thuận, 6 = Đi theo nhóm. |
| DOLocationID | Text | Mã id địa điểm trả khách. |
| PULocationID | Text | Mã id địa điểm đón khách. |

## 2010 Census Tracts

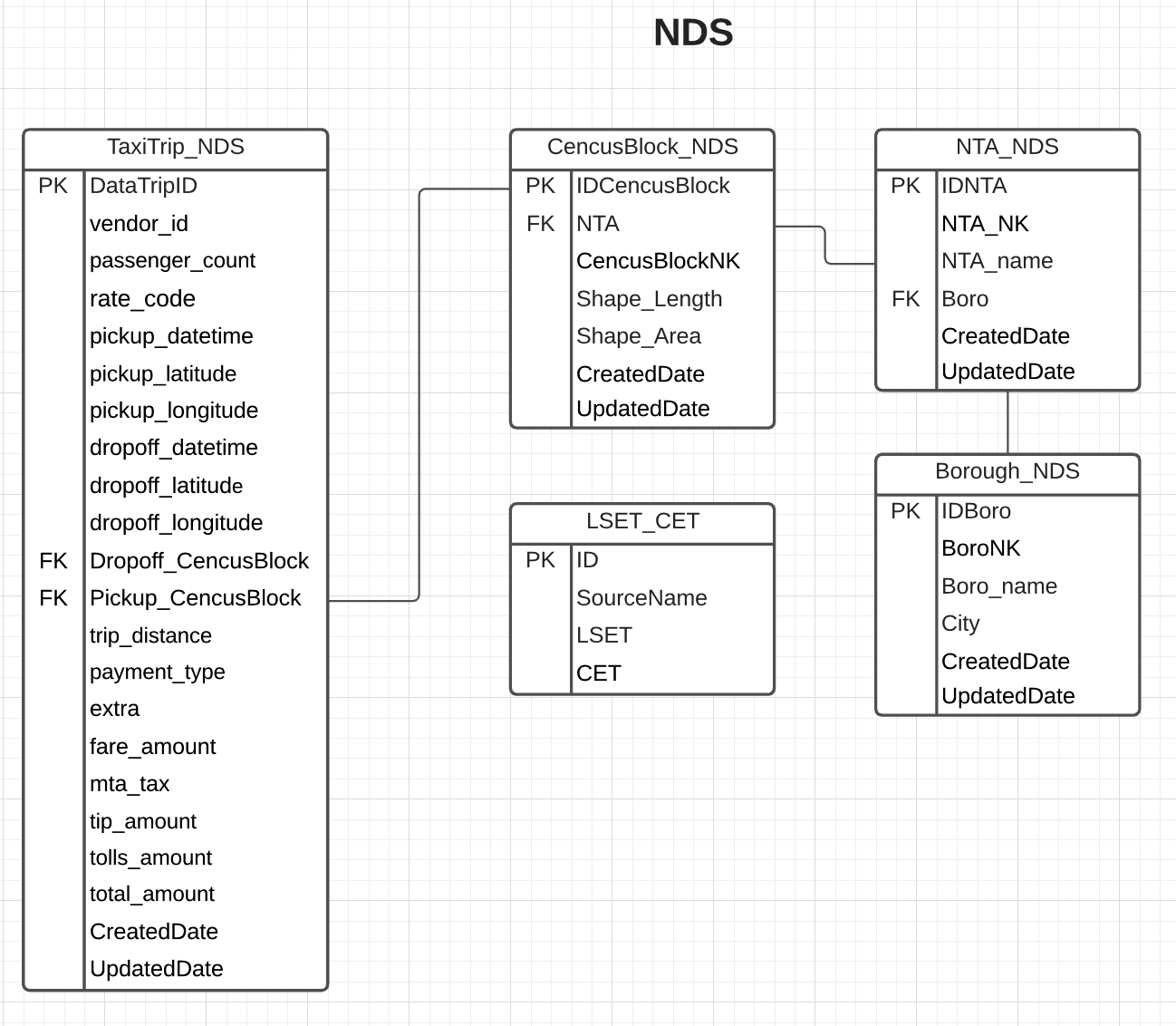
|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Description** |
| CTLabel | varchar(10) | Nhãn đánh dấu một khu vực (hình đa giác trên map).  Mỗi CTLabel là duy nhất ở quận đó (Có thể trùng nhau ở khác quận). |
| BoroName | varchar (50) | Tên của một quận ở thành phố New York.  Có 5 quận bao gồm: Manhattan, Bronx, Brooklyn, Queens, Staten Island. |
| BoroCode | varchar (10) | Mã của quận. |
| CT2010 | Varchar(6) | Dùng để làm rõ hơn thuộc tính CTLabel sẵn có.    Gồm 6 chữ.  = CTLabel x 100 |
| BoroCT2010 | Varchar(7) | Dùng để phân biệt các khu vực CT của cả thành phố NY (Mỗi khu vực sẽ có ID duy nhất).  Gồm 7 chữ số.  = BoroCode x 1 000 000 + CT2010 |
| CDEligibil | Char() | Community Development Block Grant Eligibility - Tính đủ điều kiện tài trợ của Khối Phát triển Cộng đồng.  Range Value: E - I tương yes - no |
| NTACode | Varchar(10) | Mã khu vực bảng vùng lân cận.  Chữ cái đầu là viết tắt của tên quận.  Ví dụ: Brooklyn - BK01 |
| NTAName | Varchar(50) | Tên khu vực lập bảng vùng lân cận. |
| PUMA | Varchar(10) | Khu vực vi dữ liệu sử dụng công cộng.  Là các đơn vị địa lý được Điều tra dân số Hoa Kỳ sử dụng để cung cấp thông tin thống kê và nhân khẩu học.  Mỗi quận là 2 chữ số đầu khác nhau.  Ví dụ: Bronx - 3701, Manhattan - 3801,.. |
| Shape\_Leng | Float | Chu vi một CT. |
| Shape\_Area | Float | Diện tích một CT. |

# Data Warehouse Design

## Stage Design



## NDS Design



## DDS Design

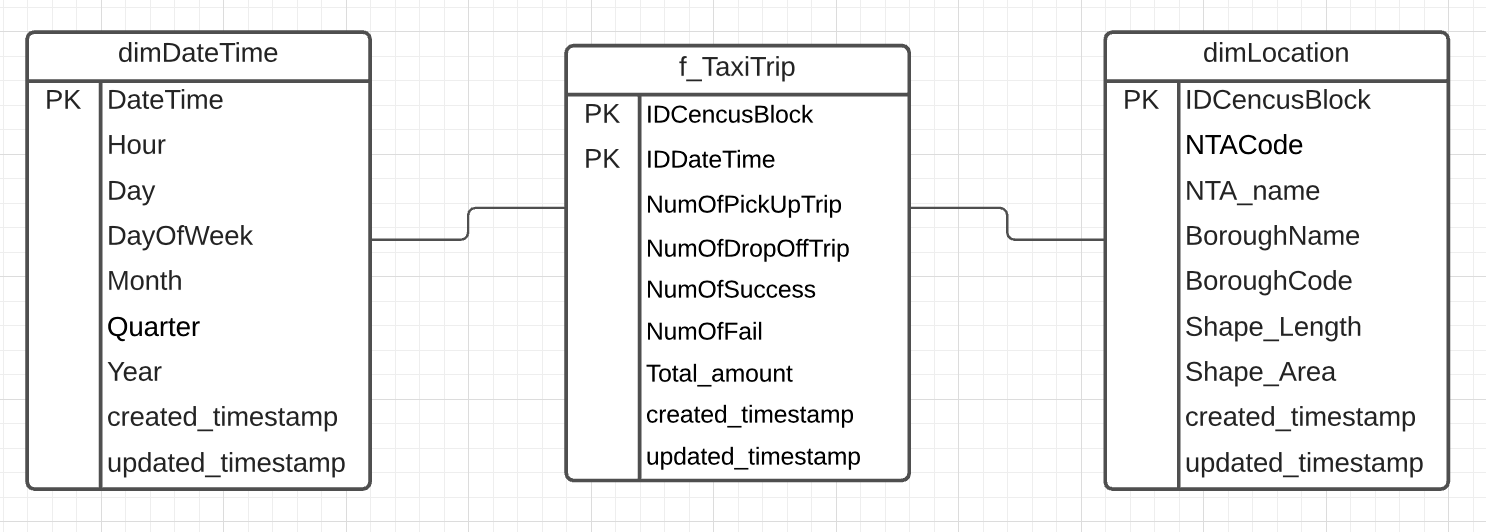
**Phân tích yêu cầu:**

**Sự kiện:** Khi một kết thúc một quy trình đặt xe

**Bối cảnh sự kiện:**

* Địa điểm: Location - địa điểm đưa đón của taxi
* Khi nào: DateTime - thời gian đưa đón của taxi.

**Đo lường**: total\_amount, NumOfPickUpTrip, NumOfDropOffTrip,  
NumOfSuccess, NumOfFail



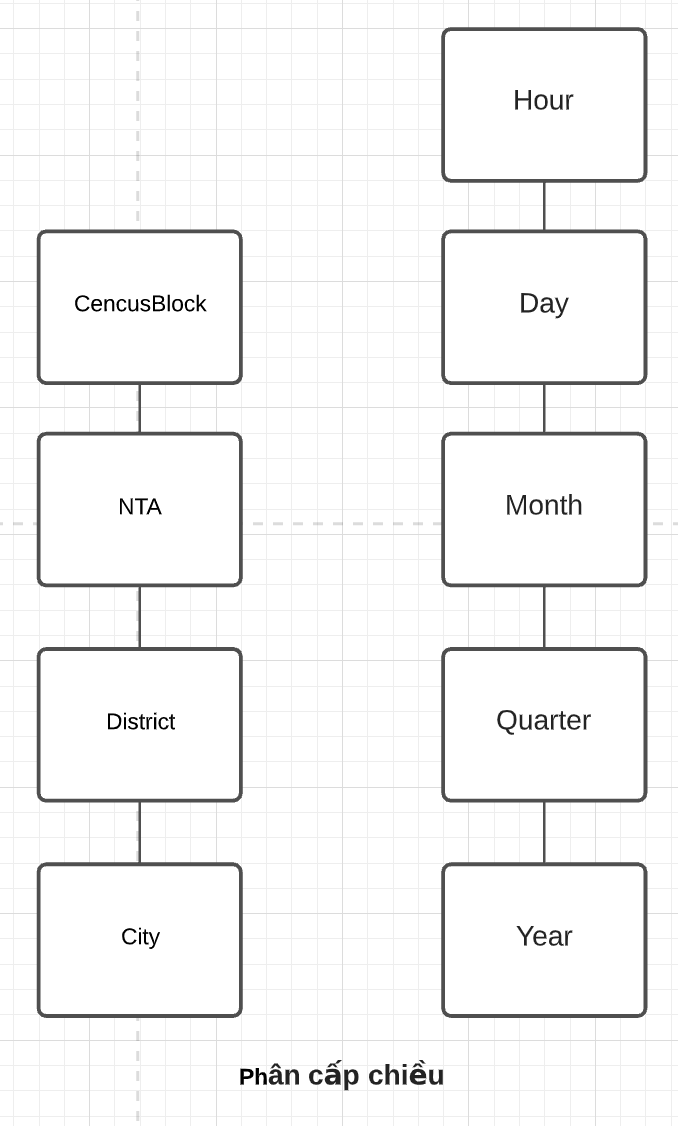
**Fact table:**

* Các giá trị có sẵn từ nguồn: total\_amount.
* Các giá trị phải tính toán:
  + NumOfPickUpTrip: tổng chuyến taxi được đặt theo giờ tại một khu vực Census Block.  
    Dùng count(\*) theo giờ và khu vực
  + NumOfDropOffTrip: tổng chuyến taxi đến nơi theo giờ tại một khu vực Census Block.  
    Dùng count(\*) theo giờ và khu vực
  + NumOfSucess: Tổng số chuyến taxi thành công tại thời điểm drop off
  + NumOfFail: Tổng số chuyến taxi thất bại tại thời điểm drop off
* Cấp chi tiết dữ liệu (độ mịn):
  + Đơn vị nhỏ nhất xảy ra sự kiện: ***Một dòng trong Fact tương ứng tất cả chuyến taxi (đặt xe/dừng xe) tại một Census Block trong một giờ.***

**Dim Table:**

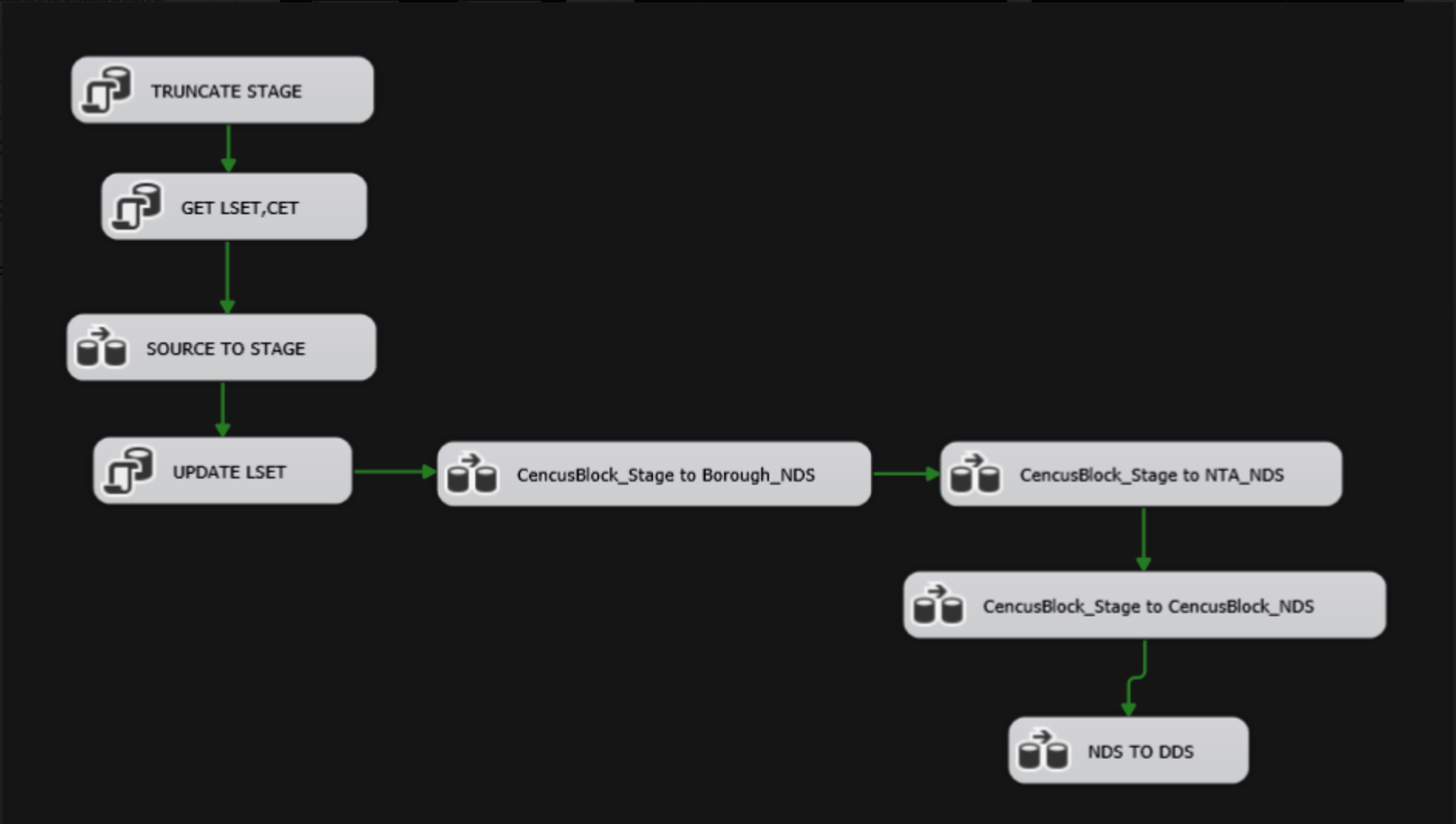
* dimDateTime: Không yêu cầu lưu giá trị cũ -> SCD 1.
* dimLocation: Không yêu cầu lưu giá trị cũ -> SCD 1.

**Phân cấp chiều của Dim:**

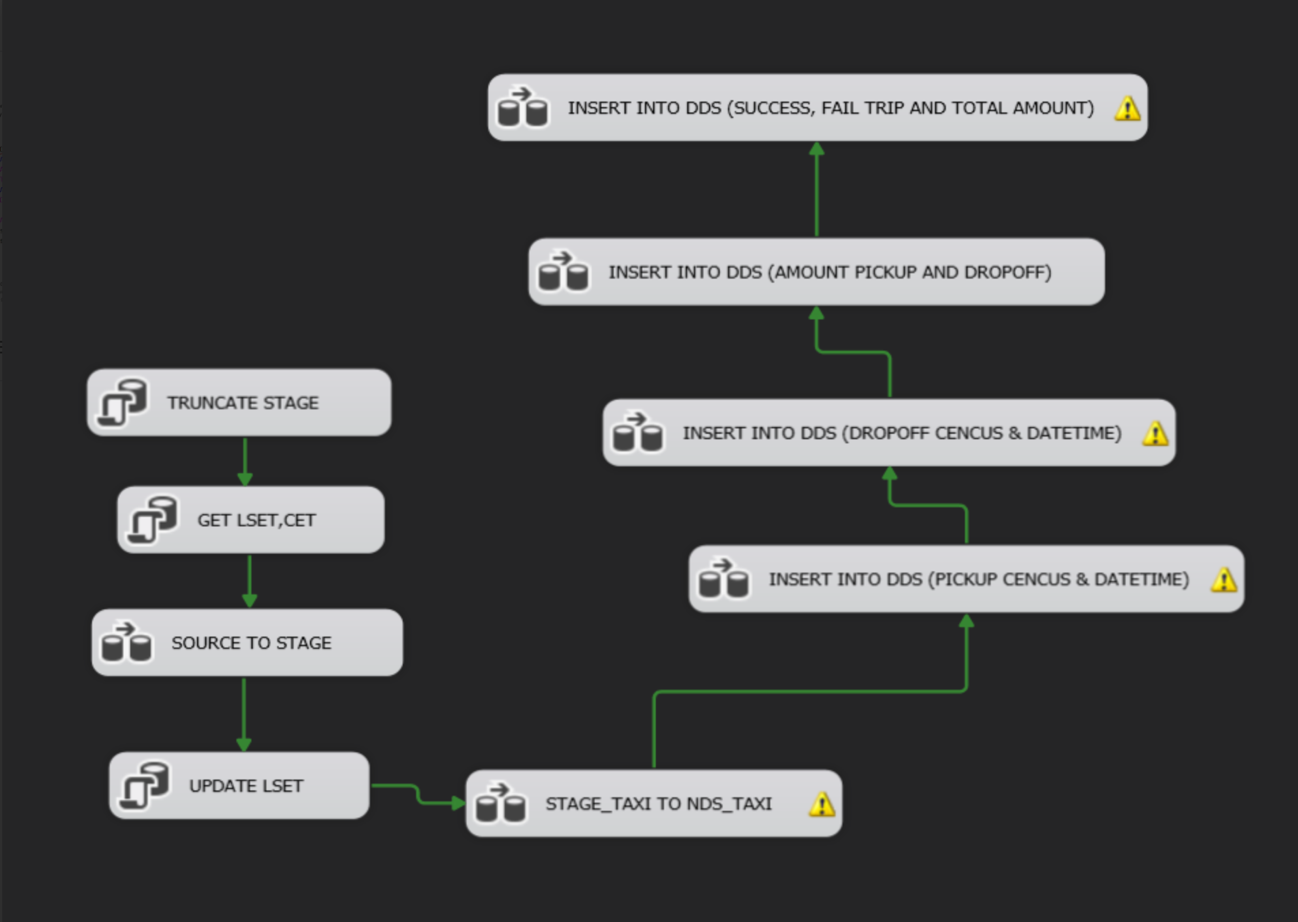


## Data Flow Design

* Source CencusTracts to DDS

****

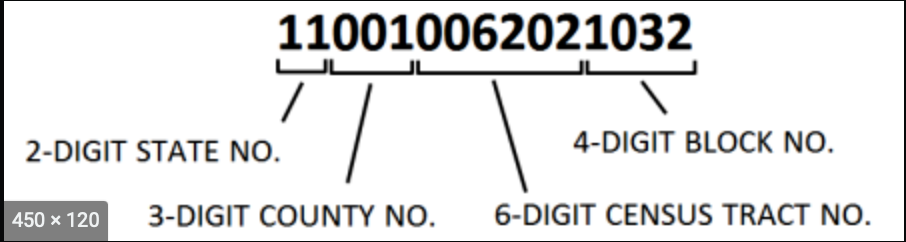
* Source Yellow Taxi Data Trip to DDS

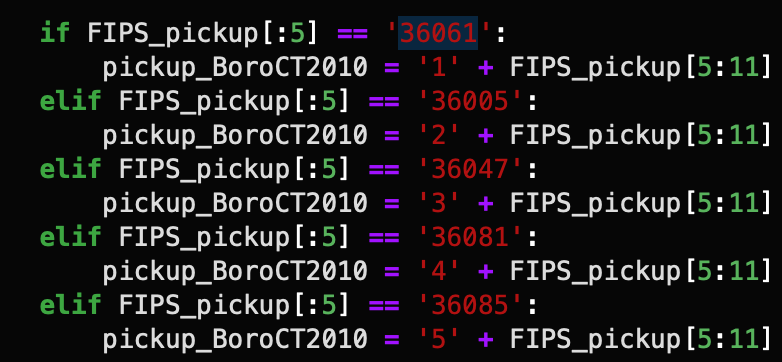


# Data Preparation and Processing

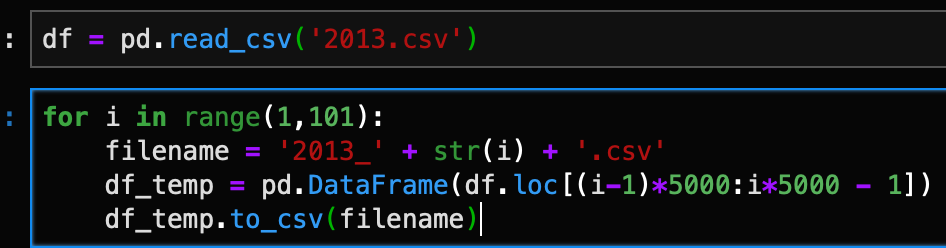
* Mỗi dòng dữ liệu ở source “Taxi Trip” là một chuyến đi được thực hiện và có lưu lại tọa độ pick up/ drop off. Còn ở source “CencusTracts” là những block và định danh của nó.
* Như ở trên thiết kế NDS cho TaxiTrip\_NDS, em đã thiết kế 2 thuộc tính Pickup\_CencusBlock và Dropoff\_CencusBlock để có thể cho biết được chuyến đi đó được pick up/drop off ở Cencus Block nào.
* Em đã tìm được một web api để lấy dữ liệu CencusBlock ở NYC dựa trên tọa độ. ([https://geo.fcc.gov/api/census/area?lat={}&lon={}&format=json](https://geo.fcc.gov/api/census/area?lat=%7b%7d&lon=%7b%7d&format=json))
* API này trả về kết quả như sau:

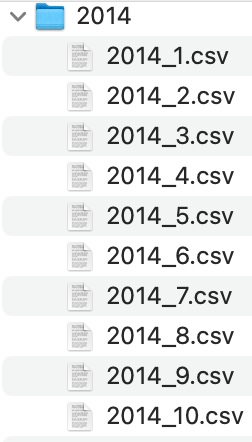


* Sau khi tìm hiểu thì tụi em biết được “block\_fips” là một cái code biển diễn như sau:  
  
* 5 số đầu biển diễn cho state và county và 6 chữ số sau đó biểu diễn cencus block ID. Vì thế nên em viết một hàm python chuyển đổi như sau:

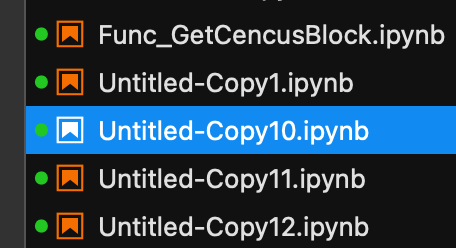


* Như vậy là em có thể lấy được CencusBlockID từ tọa độ. (Chi tiết hơn tụi em đã up file code ở trên drive với tên **Func\_GetCencusBlock.ipynb**)
* Nhưng có một vấn đề là cần 2s xử lý với mỗi tọa độ để nhận về được cencusblockID mà dữ liệu lên đến 1,5 triệu dòng => Cần 3 triệu giây để xử lý
* Giải pháp của tụi em là sẽ tách file 1,5 triệu dòng thành 300 file csv với mỗi file chứ 5000 dòng dữ liệu để thực hiện chạy song song.





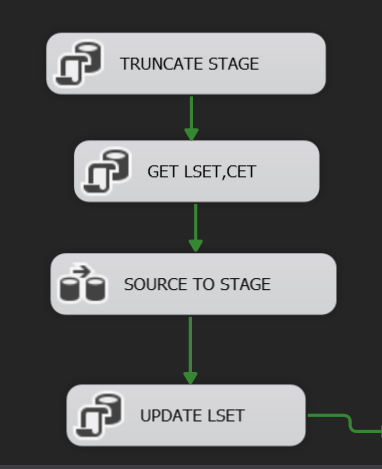
* Tụi em duplicate file **Func\_GetCencusBlock.ipynb** thành 25 file, với mỗi file sẽ chạy mỗi file excel đã tách rồi thực hiện chạy tất cả 25 file song song.



* Với giải pháp như vậy, tụi em đã giảm từ 3 triệu giây => 120 000s ~ 1,39 ngày. Có thể tăng thêm số file chạy song song để giảm số giây.
* Sau khi có được pickup\_CencusBlock và dropoff\_CencusBlock ở file csv, tụi em tiến hành đưa vào DW.

# ETL

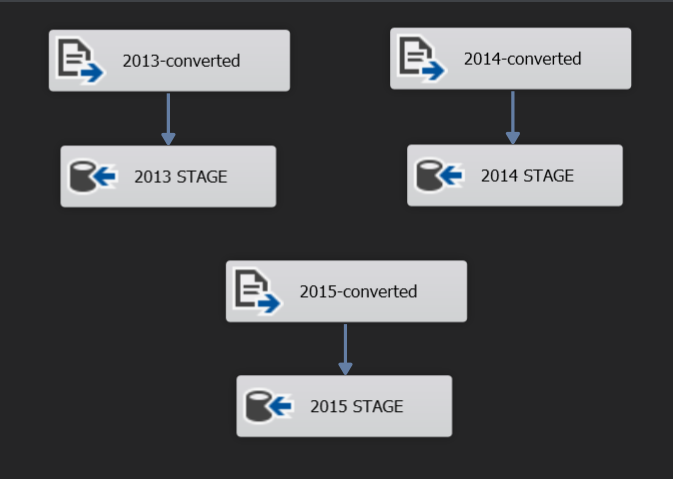
## Source to Stage



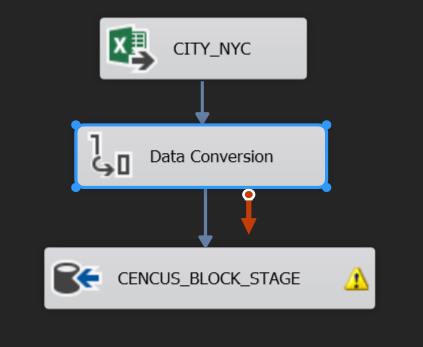
Đầu tiên, trước khi đổ dữ liệu từ nguồn vào Stage, ta cần truncate Stage trước để chắc chắn dữ liệu đổ vào không bị lặp. Sau đó, thực hiện lấy ngày LSET, CET để gán làm cột mốc cho các dữ liệu được đổ vào.

Tiếp theo, tại Source to Stage của Yellow Taxi:

Do sau khi xử lý dữ liệu, ta được 3 file .csv của các năm 2013 – 2014 – 2015 nên loại Source đưa vào chọn là Flat File, sau đó map vào các bảng Stage đã được tạo sẵn trong Database.

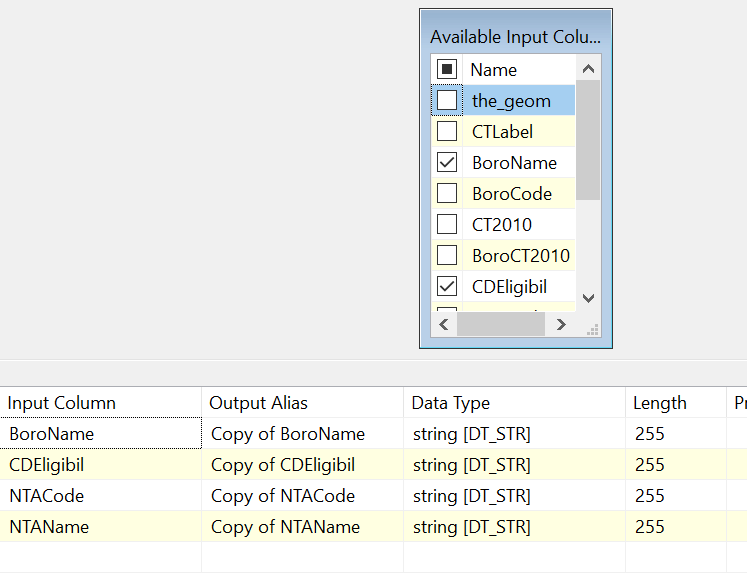


Tại Source to Stage của CencusBlock:



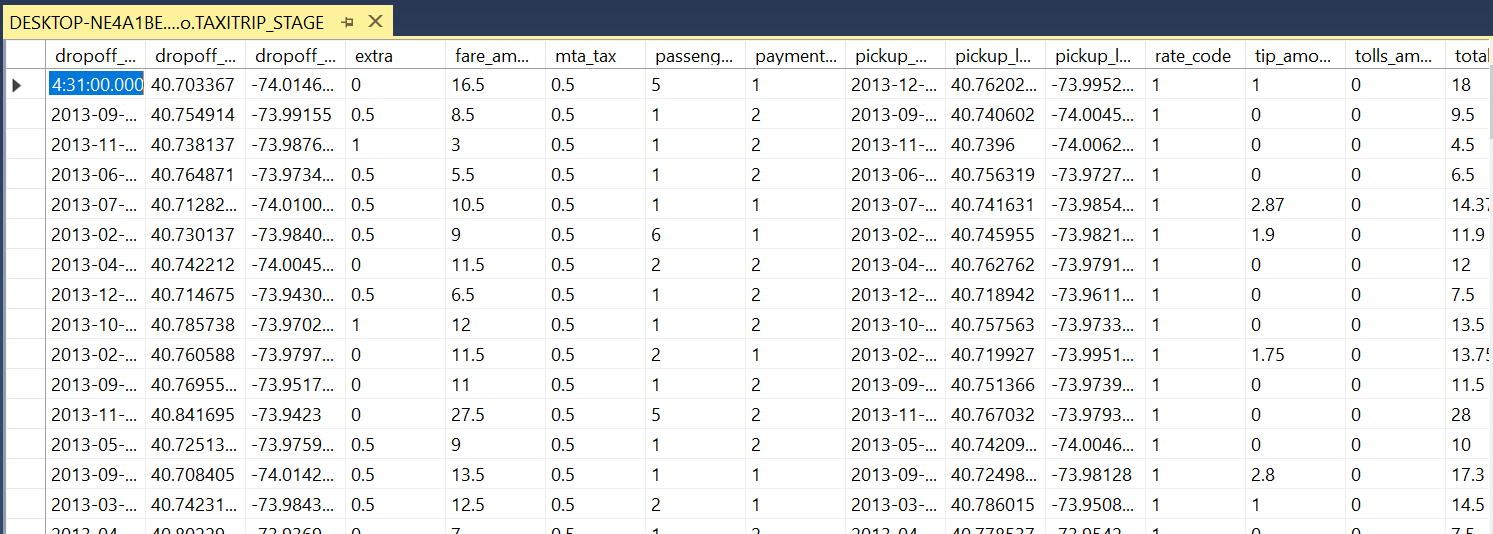
Sau khi xử lý dữ liệu, ta được file .csv City\_NYC, loại Source đưa vào có thể là Flat File hoặc là Excel. Tuy nhiên, nếu sử dụng Excel để đưa vào thì cần thêm một bước chuyển đổi dữ liệu các cột như sau:

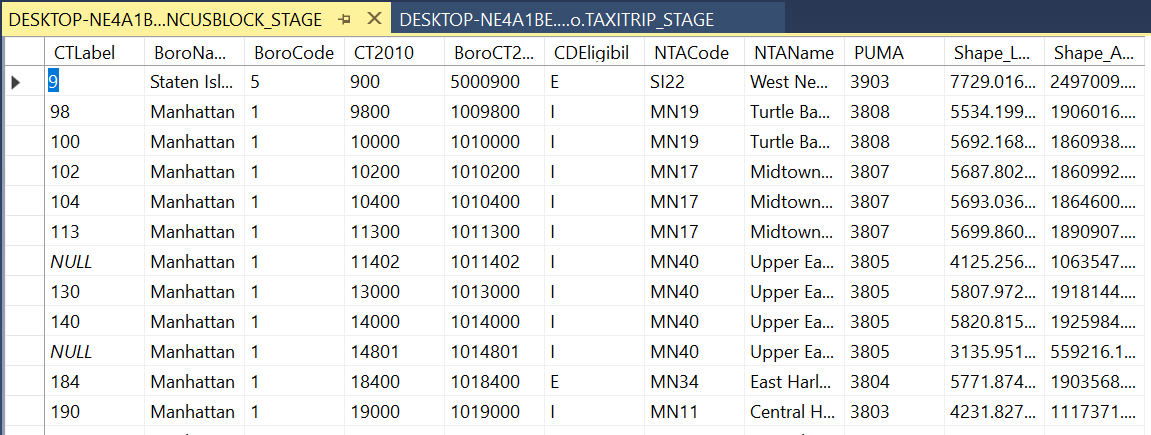
Sau đó, ta thực hiện đổ vào Stage với những dữ liệu có sẵn trong Excel và những dữ liệu vừa được chuyển đổi.



Cuối cùng, ta cập nhật lại LSET.

Kết quả như sau:





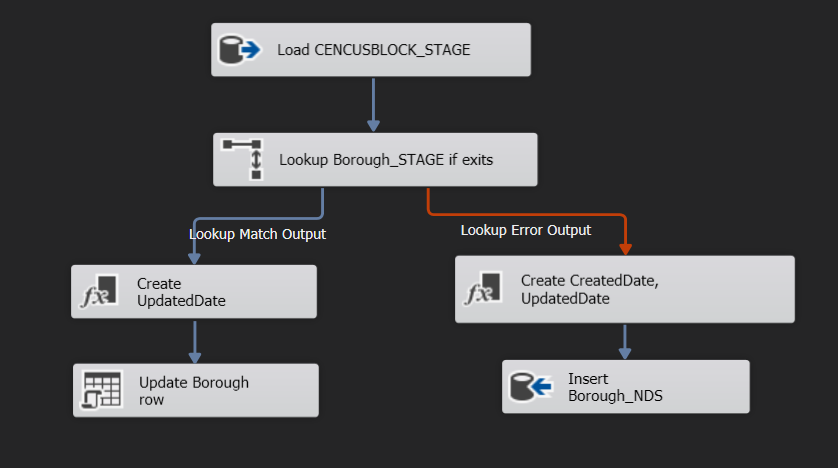
## Stage to NDS

### **NDS Cencus Block**

Theo thiết kế của NDS Cencus Block bên trên, ta chia ra thành 3 bảng tương ứng với sự phân chia từ nhỏ đến bé: Borough – NTA - CencusBlock.



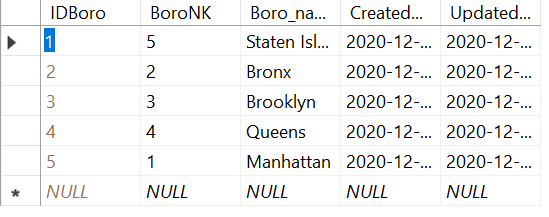
1. **CencusBlock\_Stage to Borough\_NDS:**



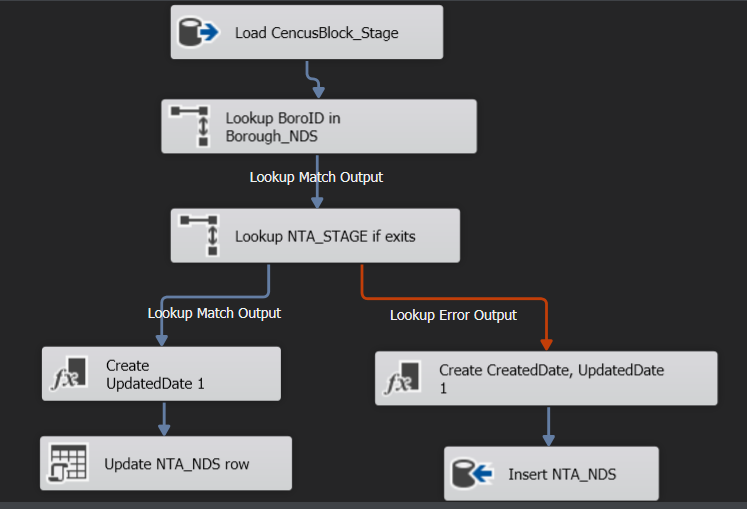
Ta làm theo trình tự:

* Load BoroCode và BoroName từ CencusBlock\_Stage
* Kiểm tra xem đã có dữ liệu Borough nào đã có trong NDS hay không:
  + Nếu có thì Update dữ liệu theo nhánh bên trái.
  + Nếu không thì Insert dữ liệu mới theo nhánh bên phải.

Kết quả như sau:

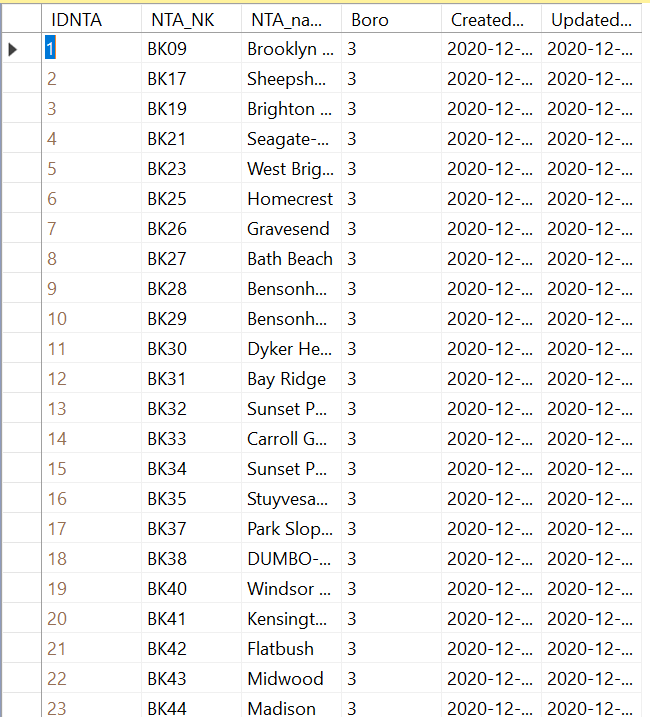


1. **CencusBlock\_Stage to NTA\_NDS**

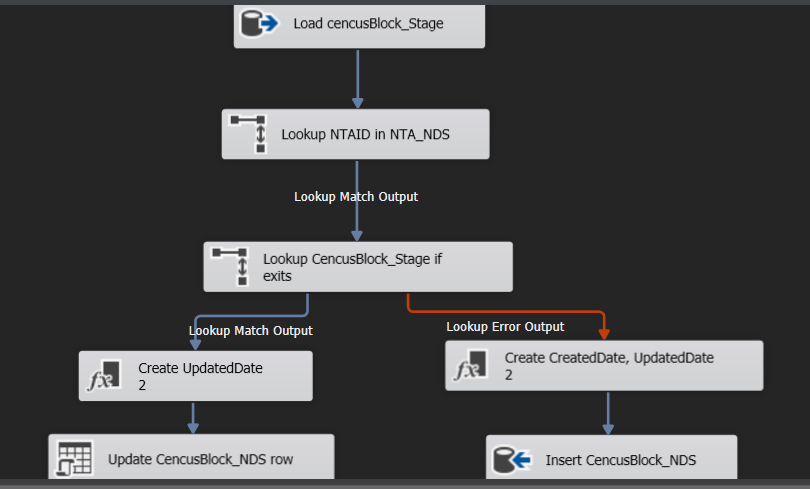


Trình tự cũng gần giống với khi đổ dữ liệu vào Borough\_NDS. Tuy nhiên, do NTA\_NDS có trỏ khóa ngoại đến Borough\_NDS nên cần kiểm tra Borough đã tồn tại trước khi kiểm tra NTA đã tồn tại. Sau đó, các bước còn lại giống như khi đổ vào Borough.

Kết quả như sau:

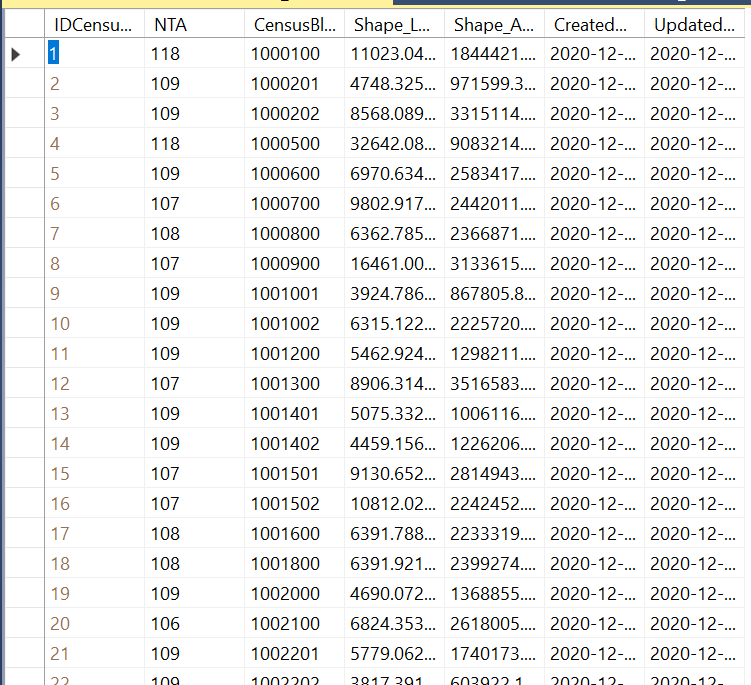


1. **CencusBlock\_Stage to CencusBlock\_NDS**

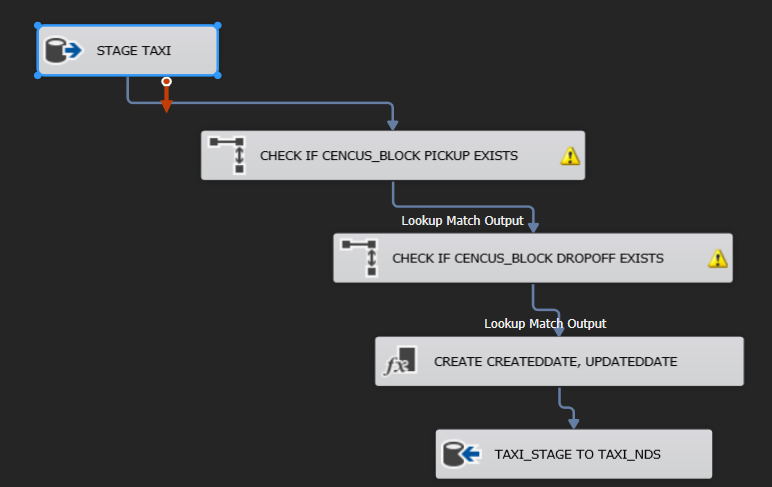


Với CencusBlock, do cũng trỏ khóa ngoại đến NTA cho nên cần phải kiểm tra NTA đã tồn tại trước. Còn lại tương tự như việc đổ dữ liệu vào NTA\_NDS như trên.

Kết quả như sau:



### **NDS Yellow Taxi Data Trip**

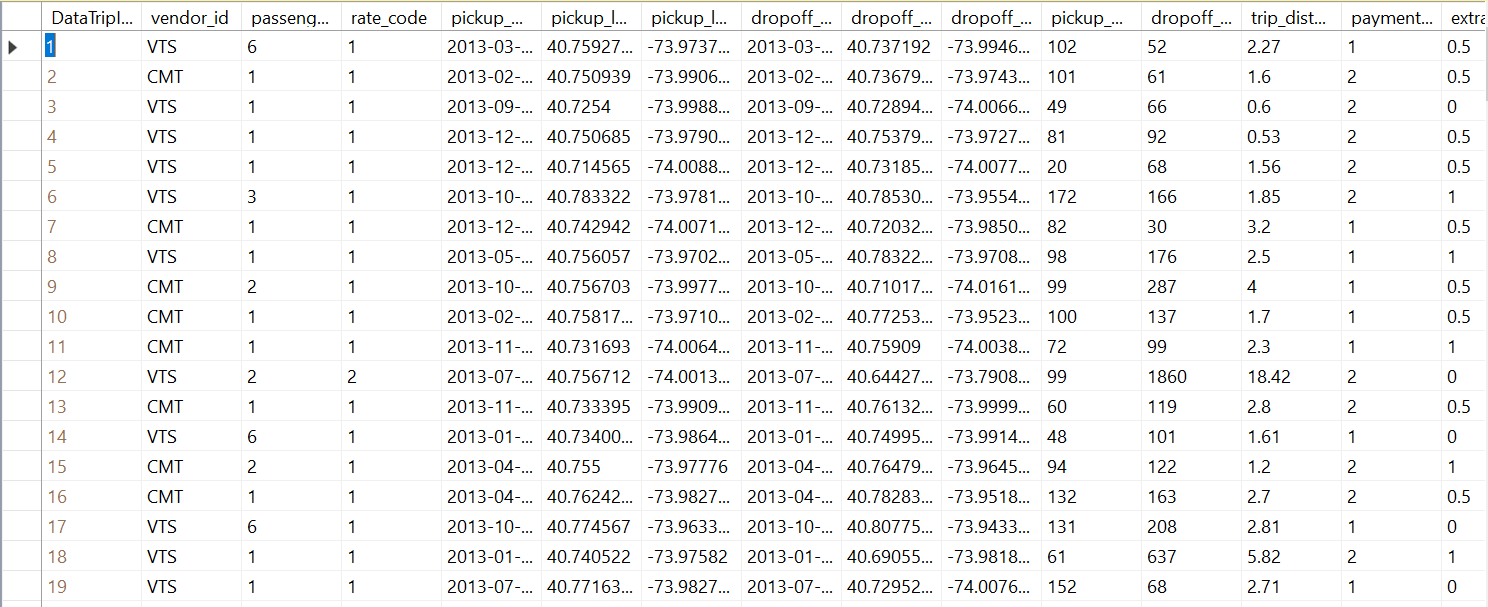


Khác với dữ liệu CencusBlock, do các dữ liệu chuyến đi Yellow Trip đều là dữ liệu lịch sử, cho nên việc đổ dữ liệu không có những thay đổi, nên không cần Update.

Đầu tiên, ta lấy dữ liệu từ Taxi\_Stage.

Tiếp theo, vì trong NDS cần lưu địa điểm đến và đi, có liên quan đến CencusBlock, nên ta cần kiểm tra dữ liệu CencusPickUp và CencusDropOff đã tồn tại hay chưa, sau đó tạo ngày và đổ dữ liệu vào NDS.

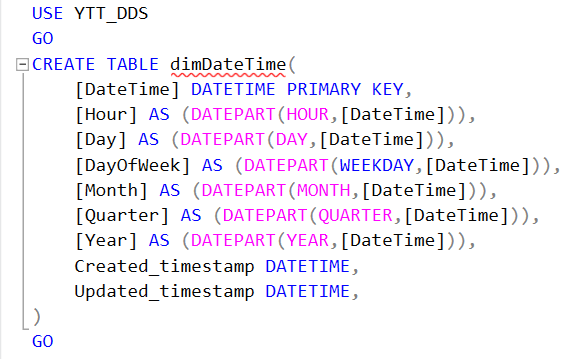
Kết quả như sau:



## NDS to DDS

### **Insert Dim DateTime**

Đầu tiên, ta tạo bảng dimDateTime trong Database DDS, khi insert dữ liệu Datetime vào thì bảng tự tách ra các thuộc tính Năm, Quý, Tháng, Ngày của tuần, Ngày, Giờ.

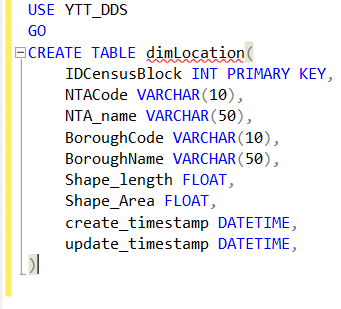
****

Sau đó, ta tiến thành insert các dữ liệu datetime vào bảng trên. Do dữ liệu nhóm thực hiện chỉ lấy trong vòng 3 năm: 2013 – 2014 – 2015 nên nhóm dùng phần mềm tự tạo dữ liệu tất cả các ngày giờ trong 3 năm đó (File dimDateTime.sql). Kết quả sau khi chạy các dòng insert thì dimDateTime được tạo thành công.

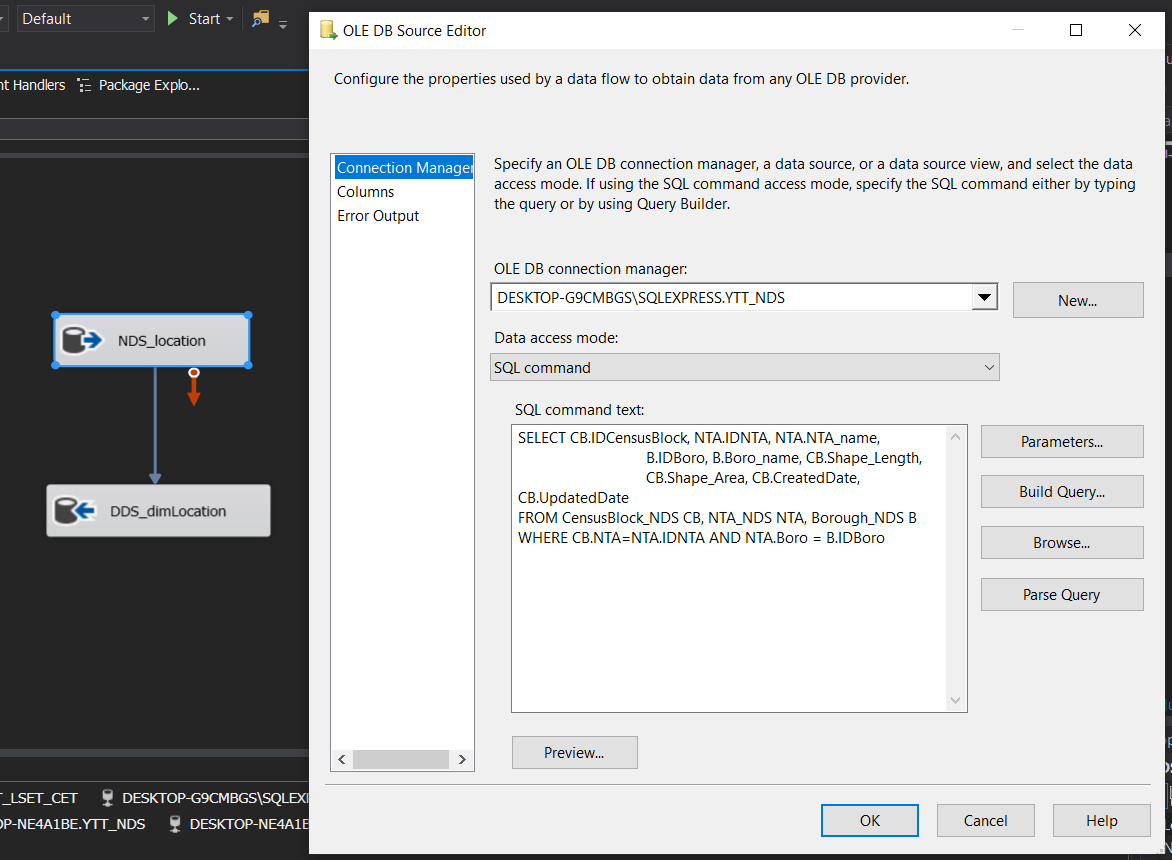
****

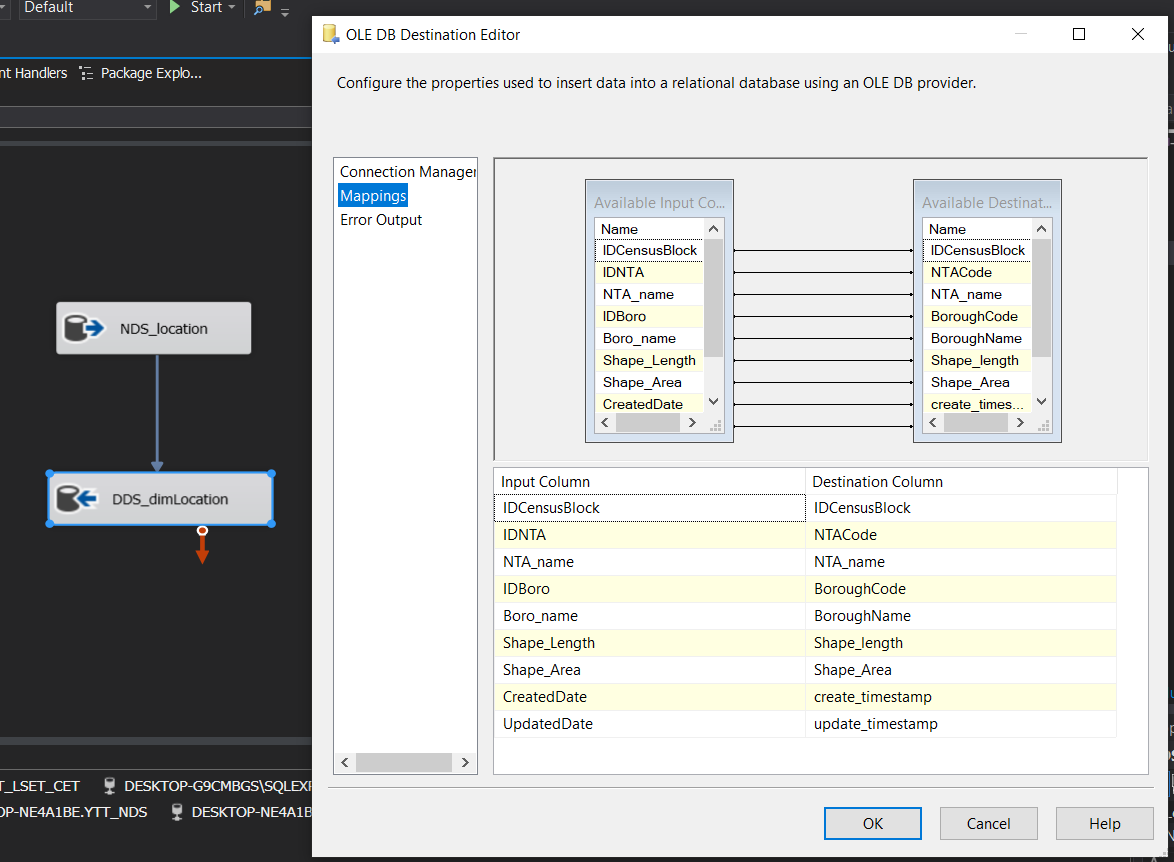
### **Insert Dim Location**

**Bước 1:** tạo bảng dimLocation

****

**Bước 2:** insert dimLocation từ NDS vào DDS

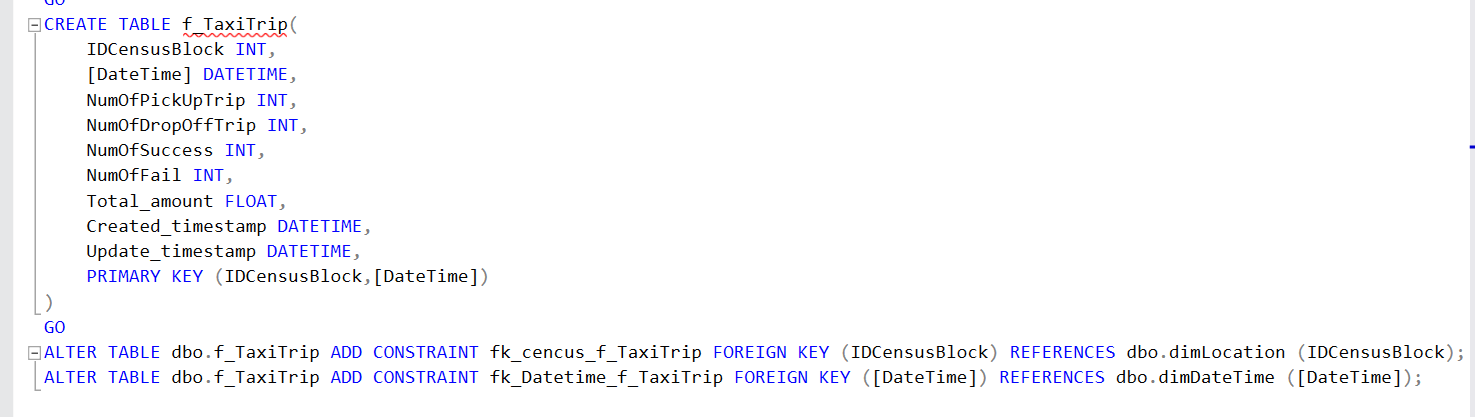
****

****

### **Insert Fact F\_TaxiTrip**

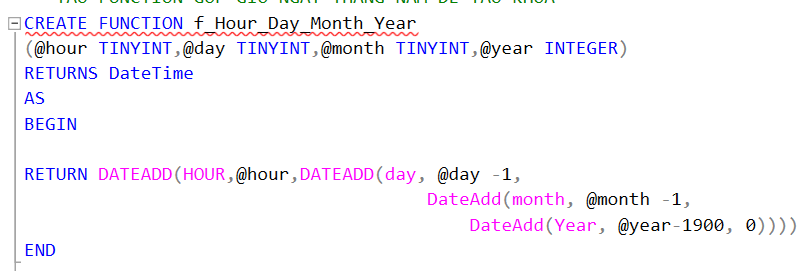
Đầu tiên, ta tạo bảng f\_TaxiTrip trong Database DDS với khóa chính của bảng chính là 2 thuộc tính IDCencusBlock và DateTime. 2 khóa chính này trỏ từ 2 thuộc tính cùng tên trong 2 Dim: Location và DateTime. Ngoài ra, bảng Fact còn chứa một số thuộc tính khác như:

* NumOfPickUpTrip: Số lượng chuyến đi đặt được
* NumOfDropOffTrip: Số lượng chuyến đi thả khách
* NumOfSuccess: Số lượng chuyến đi thành công
* NumOfFail: Số lượng chuyến đi thất bại
* Total\_amount: Tổng giá tiền chuyến đi

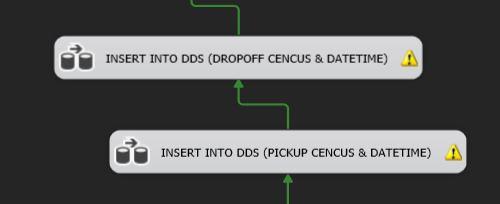
****

Tiếp theo, ta thực hiện đổ dữ liệu vào bảng trên từ NDS của Yellow Taxi.

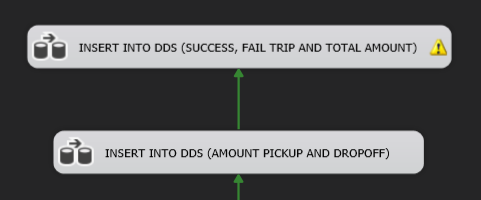
Tuy nhiên, do dữ liệu thời gian trong NDS Yellow Taxi lấy đến phút giây, mà trong DDS ta chỉ có nhu cầu lấy dữ liệu mịn đến giờ, nên ta phải tạo 1 function f\_Hour\_Day\_Month\_Year mà mỗi khi truyền dữ liệu thời gian từ NDS vào thì function sẽ chỉ lấy năm, tháng, ngày, giờ cho thuộc tính khóa DateTime**.**

****

Tiếp theo, ta đổ PickUpCencusBlock, DropOffCencusBlock với Datetime tương ứng từ NDS vào DDS và để làm khóa cho bảng Fact.

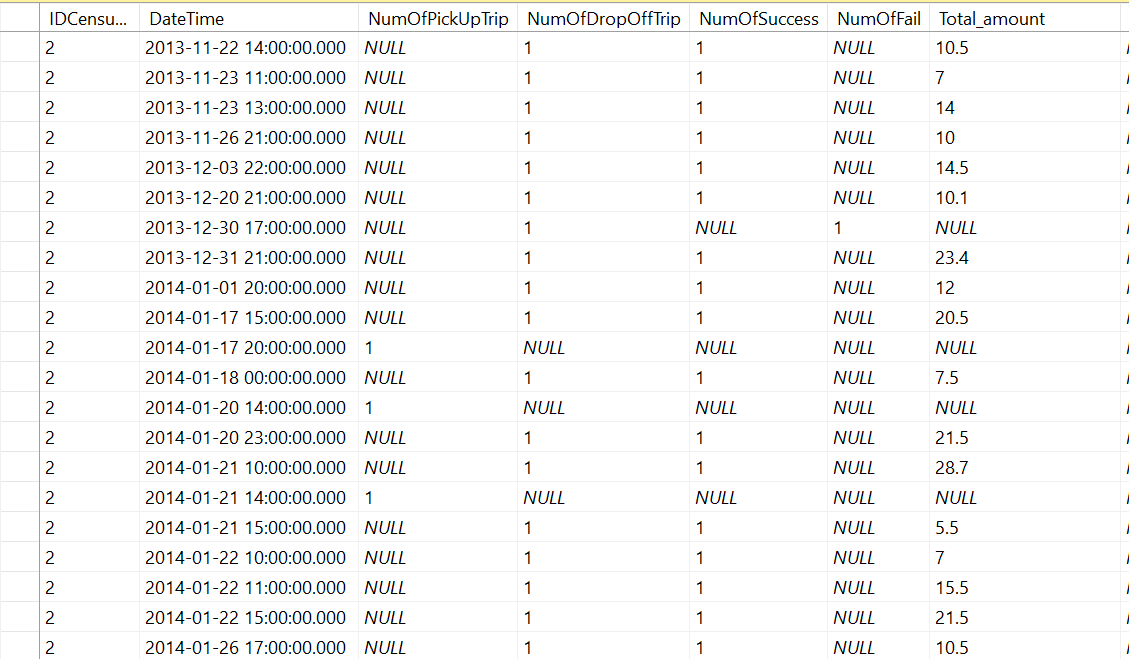


Cuối cùng, ta thực hiện đổ dữ liệu các cột AmountPickUp, AmountDropOff, SuccessTrip, FailTrip, TotalAmount vàoNDS.

****

Lưu ý: Do chỉ đổ dữ liệu đã có sẵn ở quá khứ, không có nhu cầu Update nên SCD được bỏ qua.

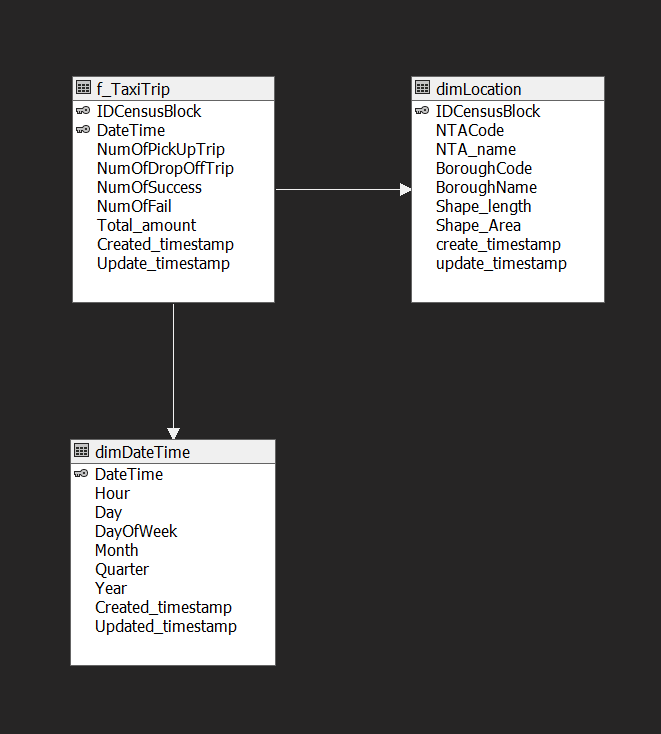
Kết quả đổ thành công dữ liệu vào bảng Fact f\_TaxiTrip.

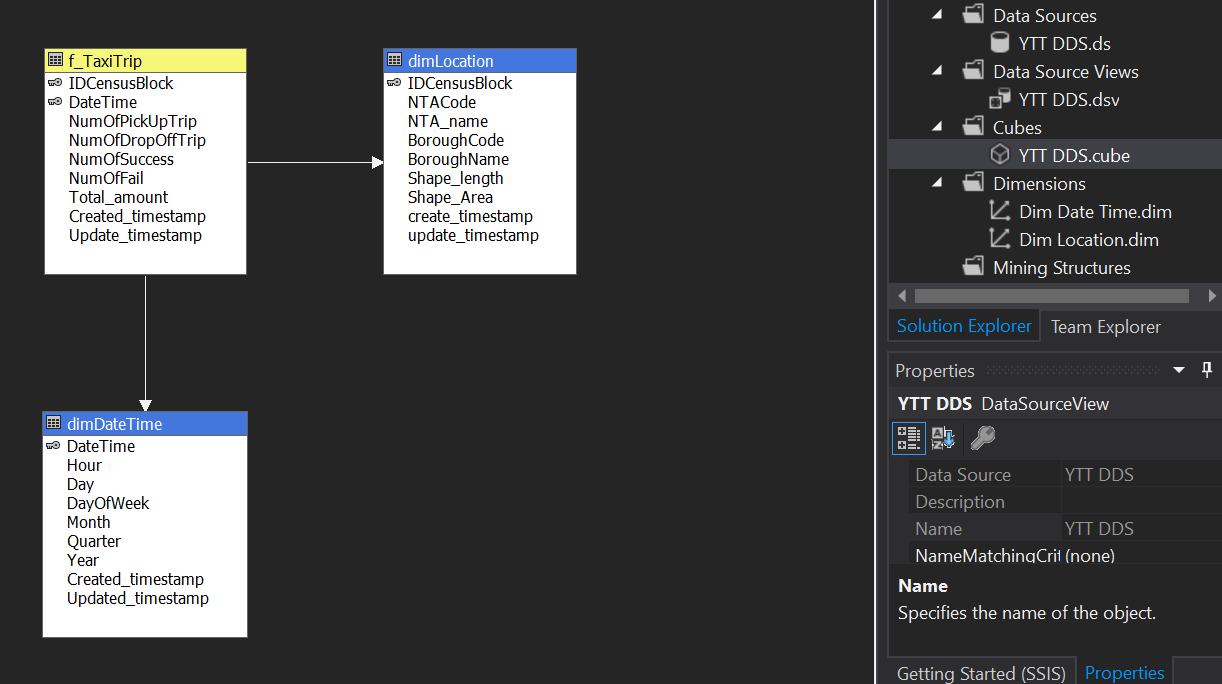
****

# OLAP

## Tạo Cube

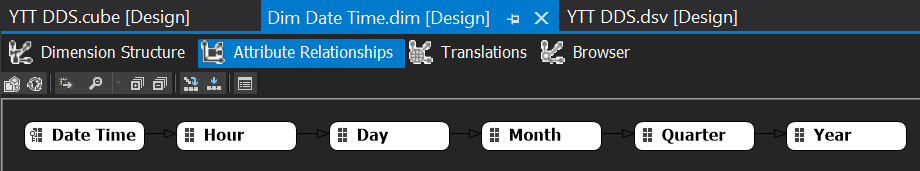
Đầu tiên, ta tạo lần lượt Data Source và Data Source View như sau:



Sau đó tạo Data Cube, chọn Measure trong f\_TaxiTrip và Dimension là DimLocation và DimDateTime  


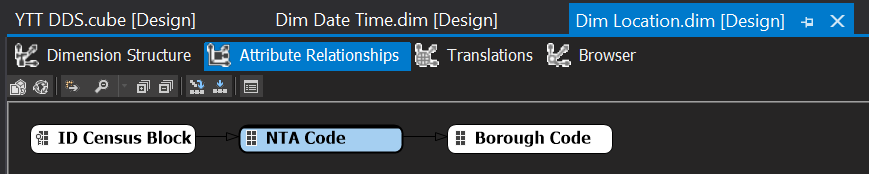
Tiếp đến, ở Dimensions, tại DimDateTime, ta phân cấp chiều theo thứ tự Năm – Quý – Tháng – Ngày – Giờ. Key được thiết lập như sau:

* Year: dimDateTime.Year
* Quarter: Collection <Year, Quarter>
* Month: Collection: <Year, Month>
* Day: Collection: <Year, Month, Day
* Hour: Collection: <Year, Month, Day, Hour>



Tương tự, thiết lập với DimLocation, ta phân cấp chiều theo Borough Code – NTA Code – ID Cencus Block. Key được thiết lập như sau:

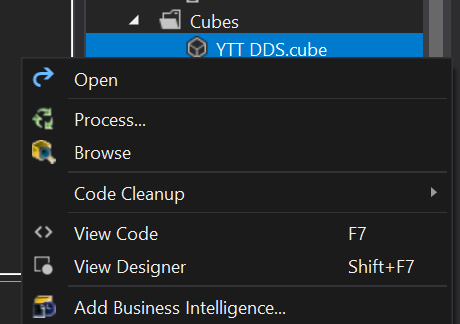
* BoroughCode: dimLocation.BoroughCode
* NTACode: dimLocation.NTACode
* IDCencusBlock: dimLocation.IDCencusBlock



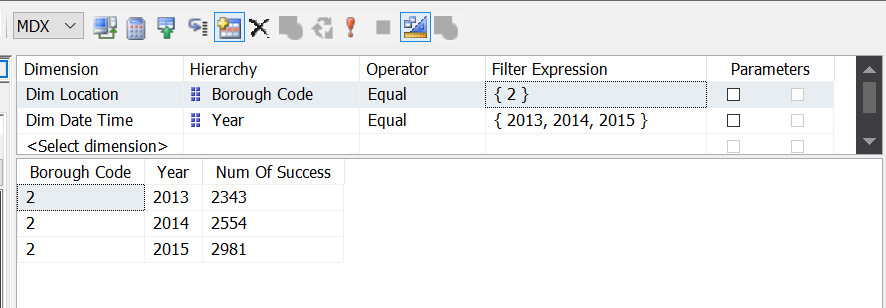
Cuối cùng, Process Cube vừa tạo.

## Browse trên Cube

Nhấp chuột trái vào Cube vừa tạo, chọn Browse.

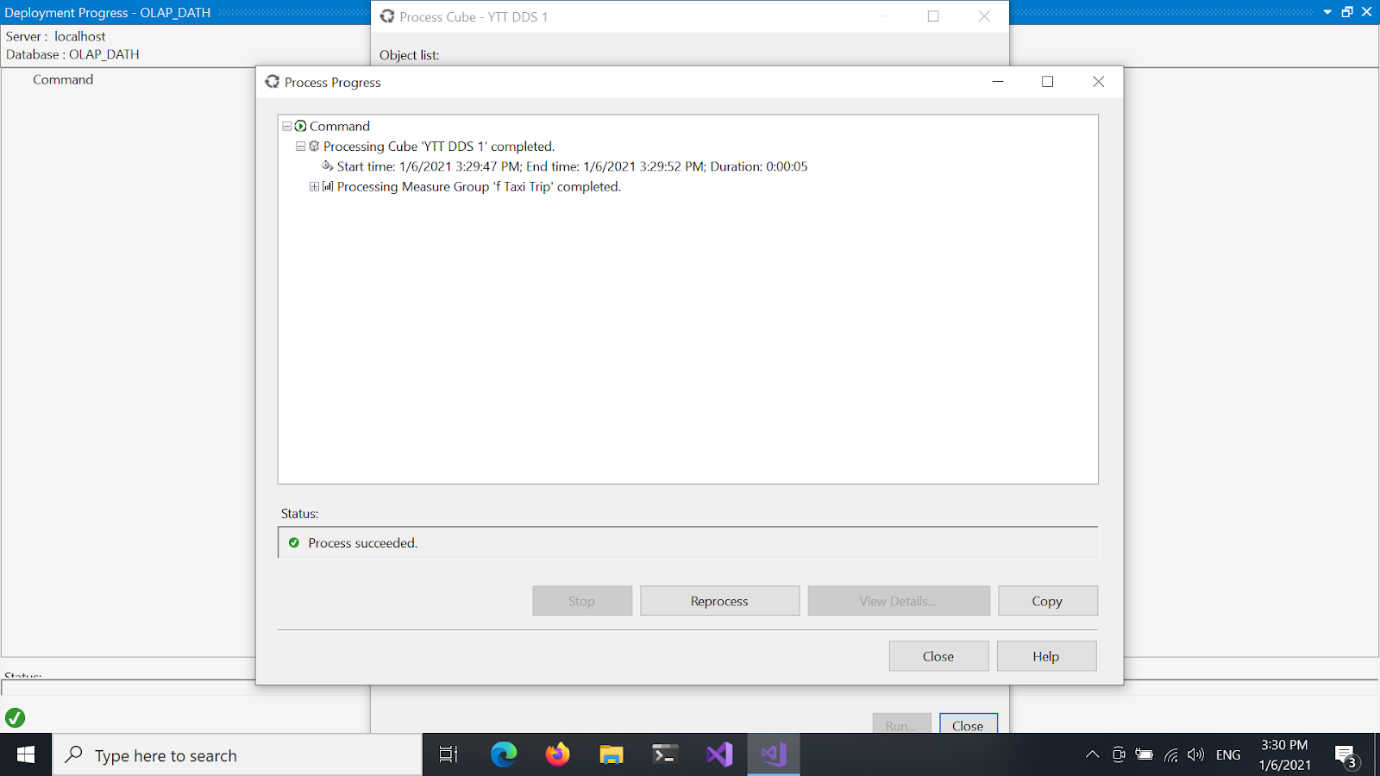


Tại đây, ta có thể kéo thả các Measure cũng như các Dim để thực hiện truy vấn nào đó. Lấy ví dụ, ta muốn xem Số lượng chuyến xe thành công tại Borough 2 theo từng năm 2013 – 2014 - 2015, trả về như sau:

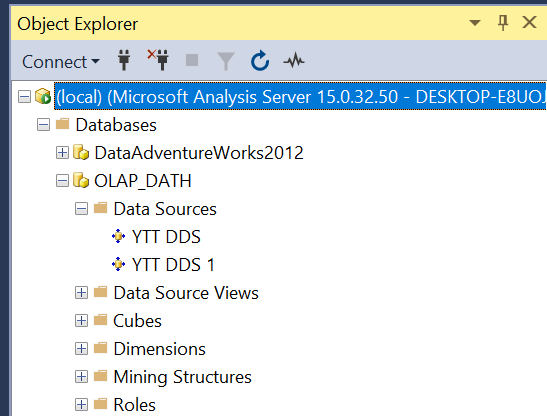


## MDX

Tiến hành deploy vào localhost để đưa vào Analysis Service Server:

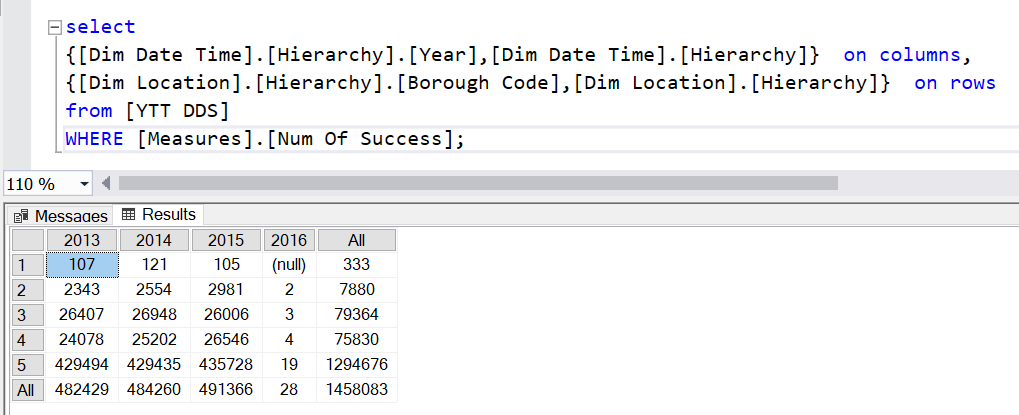


Kết quả khi vào Microsoft SQL Server (Analysis services):



Từ đây, ta có thể thực hiện truy vấn MDX.

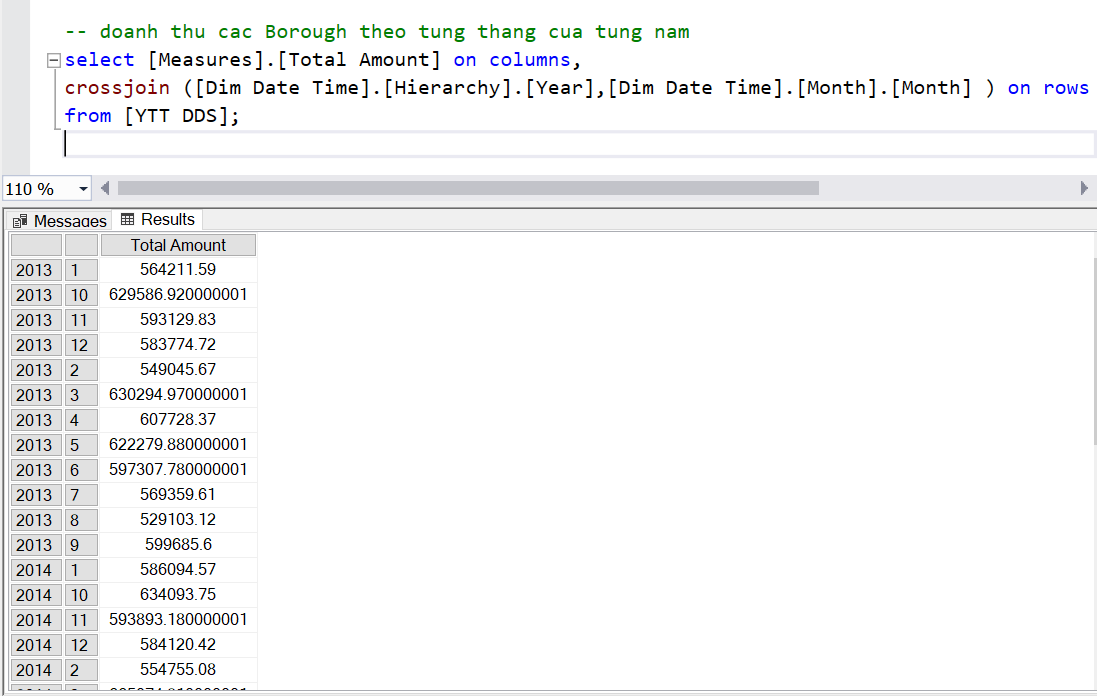
**VD1:** Tính số lượng chuyến xe thành công của từng Borough qua từng năm:



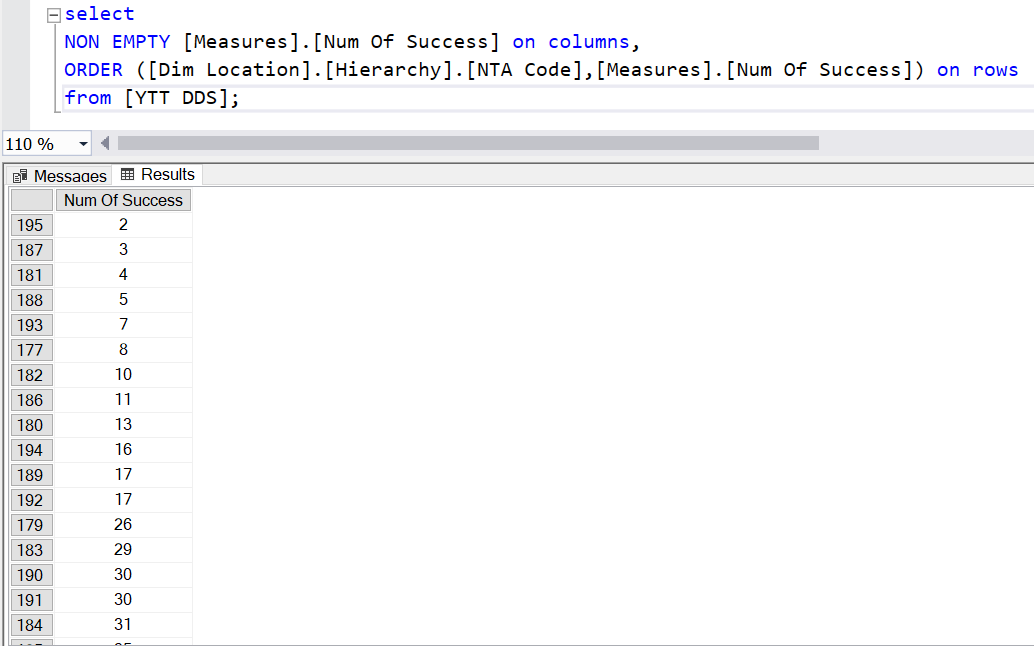
**VD2:** Doanh thu của các Borough qua từng năm:

****

**VD3:** Doanh thu của các Borough theo từng tháng của từng năm

****

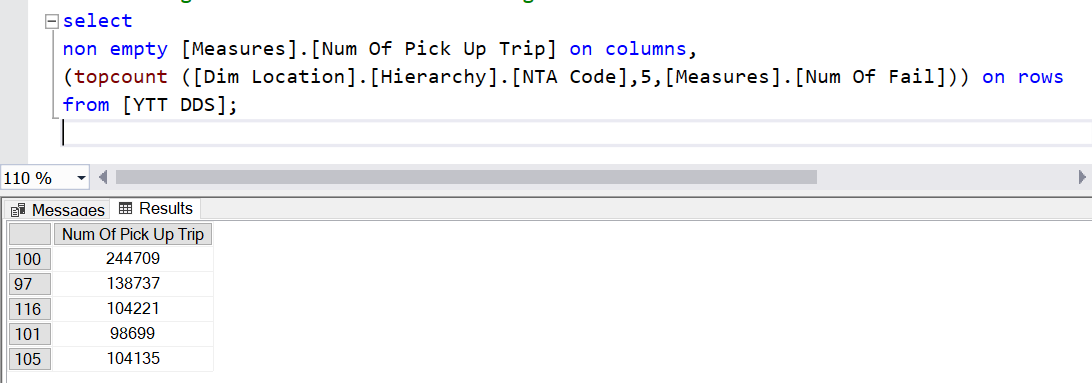
**VD4:** Sắp xếp số lượng đặt xe thành công từ thấp đến cao theo từng NTA:



**VD5:** Những NTA có doanh thu > 700000:

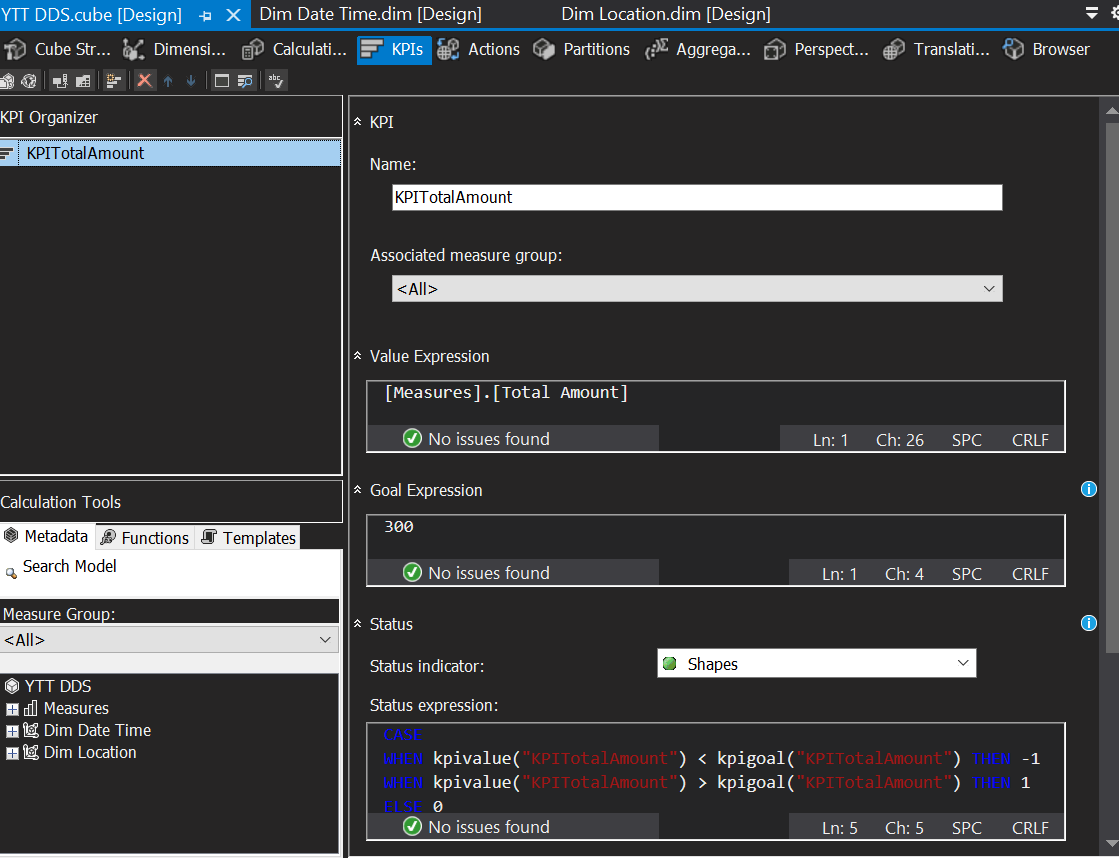


**VD6:** Số lượng đặt xe tại 5 NTA có số lượng fail cao nhất:



# KPI sass

Thiết lập KPI để kiểm tra doanh số đạt được theo từng địa điểm trên mốc thời gian với KPI = 300



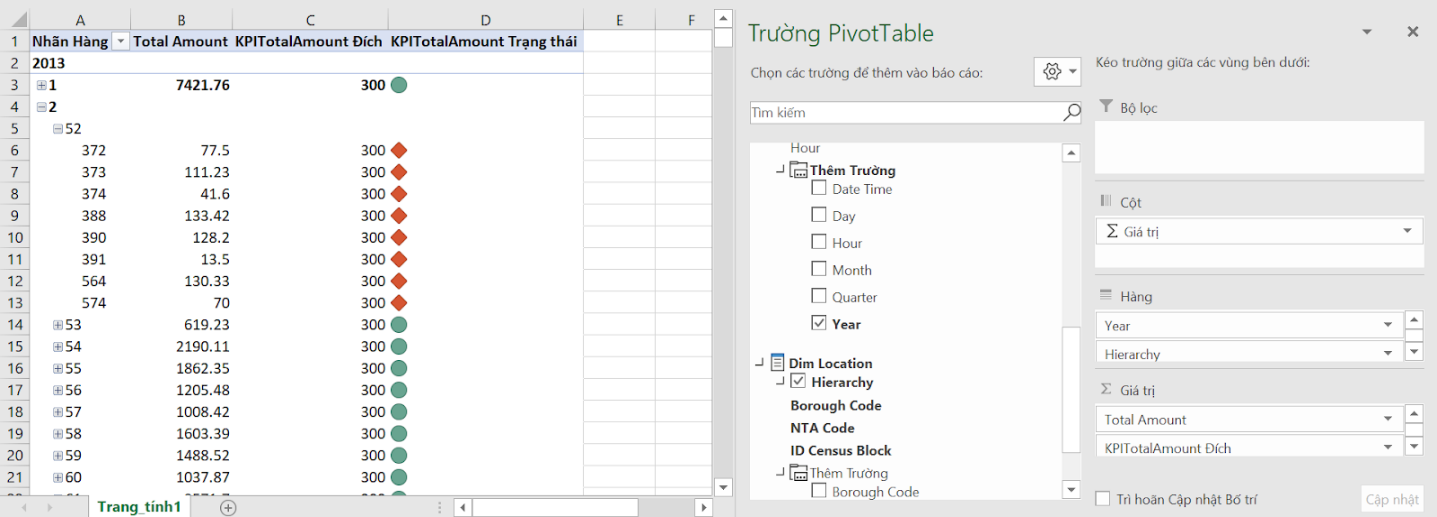
Đầu tiên, đặt tên cho KPI là “KPITotalAmount”. Tại mục Value Expression, chọn Total Amount. Sau đó chọn Goal Expression.

Để có thể trực quan giá trị nào đạt được KPI đã đề ra, tại mục Status Indicator ta chọn Shape. Để thiết lập như thế nào là đạt được KPI, ở Status Expression, ta thực hiện đoạn code như sau:

Sau đó, ta Build rồi Deploy Project OLAP\_DATH và vào ứng dụng Excel để xem thể hiện KPI.

Mở Excel, ta thực hiện kết nối đến CSDL từ Analysis Services.

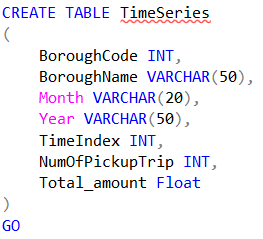
Sau đó chọn KPI (Giá trị, đích, trạng thái). Ngoài ra, tùy theo nhu cầu mà ta có thể chọn độ mịn của dữ liệu Location và Datetime để xem. Hình dưới đây thể hiện KPI Total Amount chi tiết đến từng CencusBlockID trong năm 2013.



# DATA MINING

**Mục tiêu:** Sử dụng thuật toán Time Series đểdự đoán số lượng đặt xe và doanh thu từ năm 2016 trở đi.

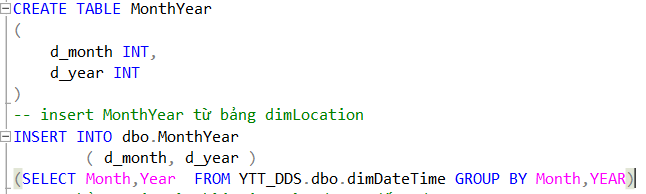
**Yêu cầu:** chuẩn bị được dữ liệu như bên dưới.



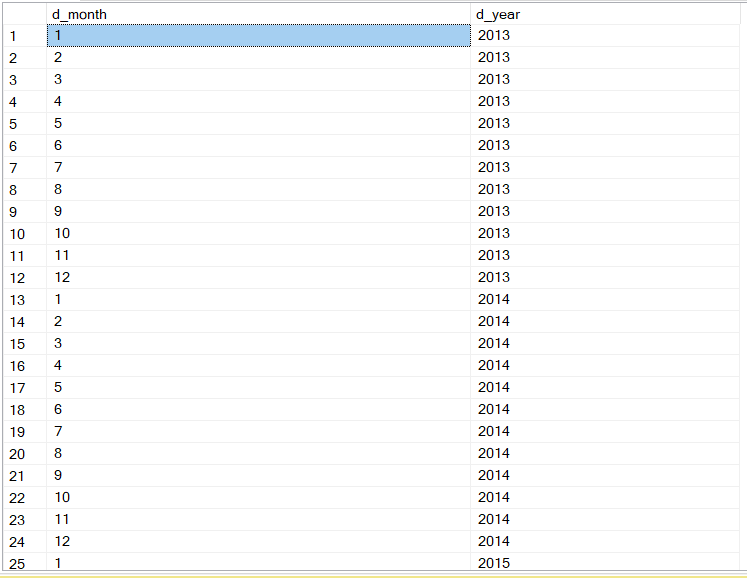
**Lưu ý:** vì dữ liệu yêu cầu không được gián đoạn nên ta thực hiện như sau

**Thực hiện chuẩn bị dữ liệu cho mining:**

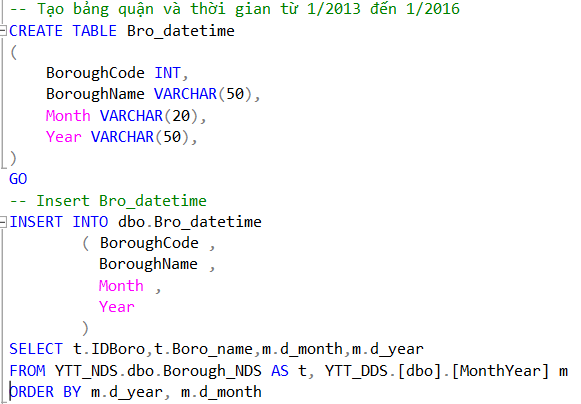
**Bước 1:** Tạo và insert bảng thời gian liên tục từ ngày 1/2013 đến 1/2016 như sau



**Kết quả:**



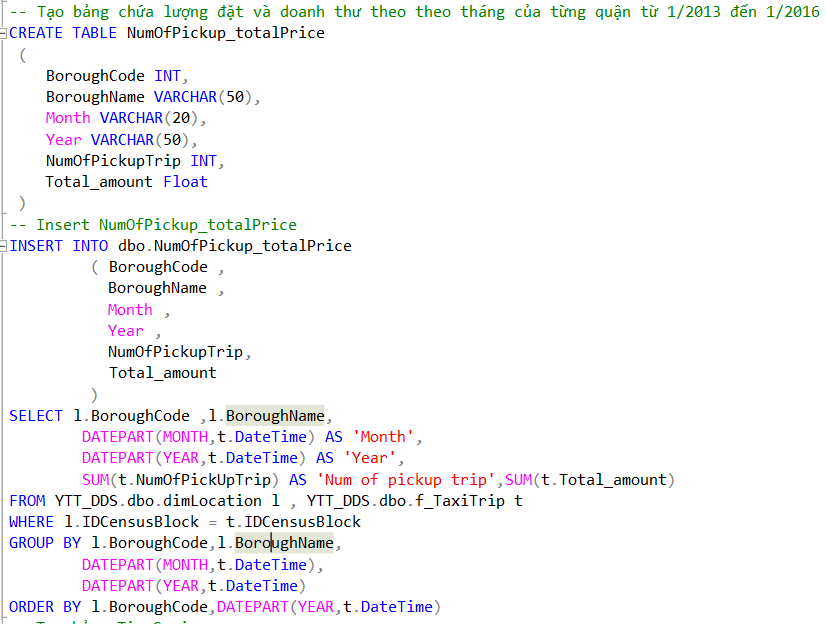
**Bước 2:** tạo và insert bảng gồm quận và thời gian từ 1/2013 đến 1/2016 như sau



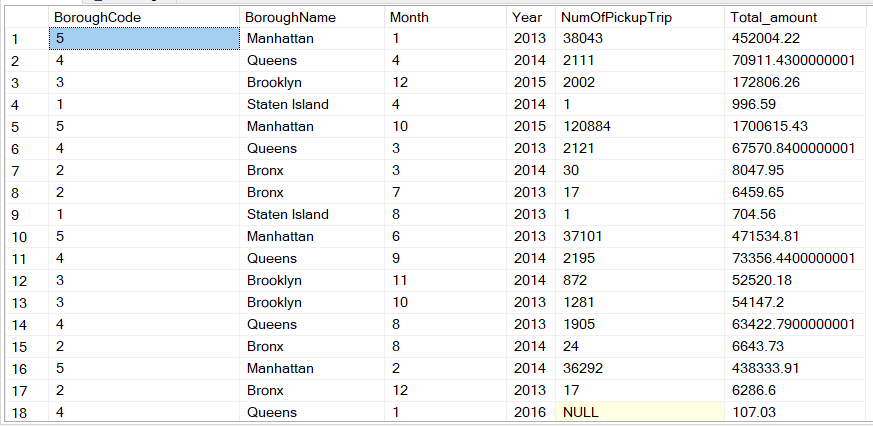
**Kết quả:**



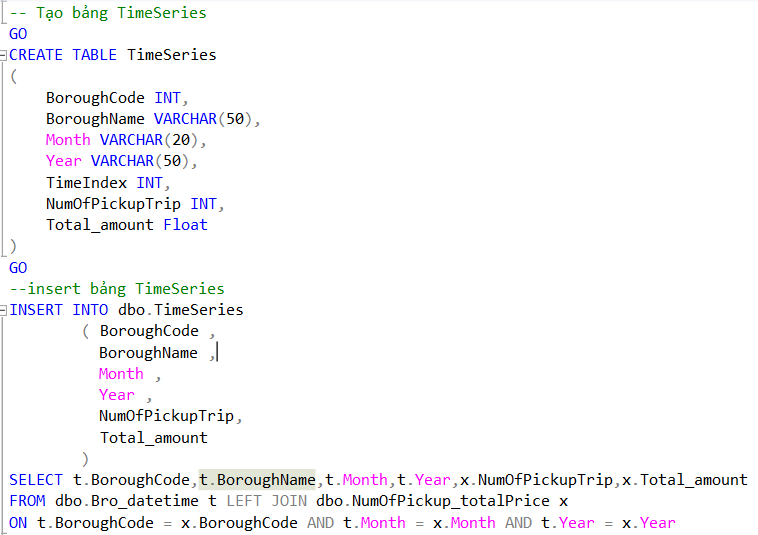
**Bước 3:** Tạo bảng chứa lượng đặt xe và doanh thu theo tháng của quận từ 1/2013 đến 1/2016



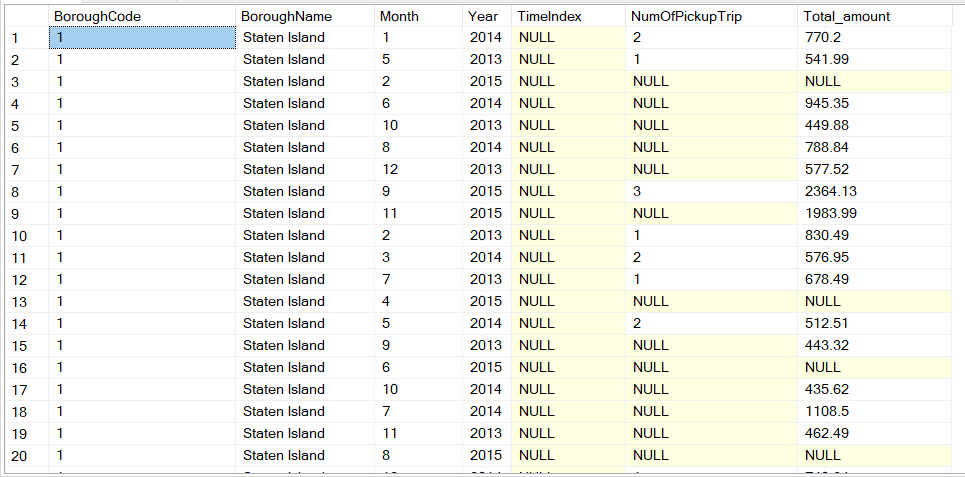
**Kết quả:**



**Bước 4:** Tạo bảng TimeSeries



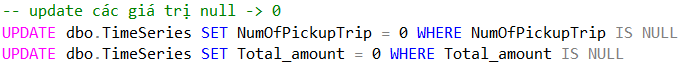
**Kết quả:**



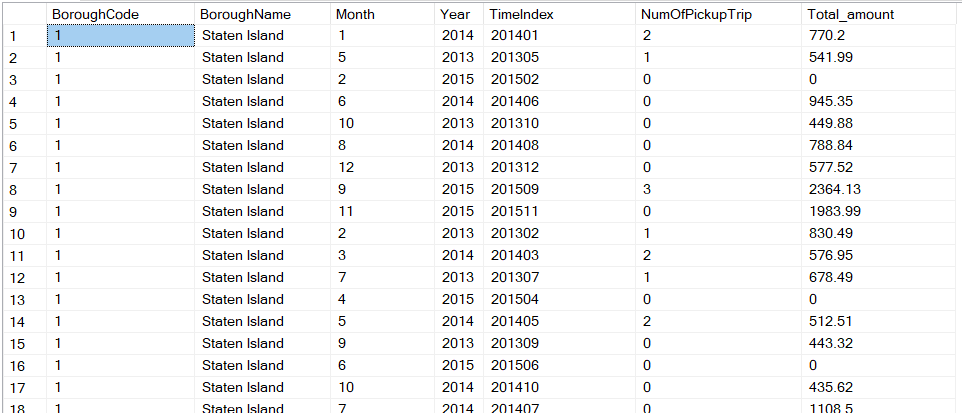
Cập nhật cột TimeIndex:



Chuyển các giá trị null thành 0:



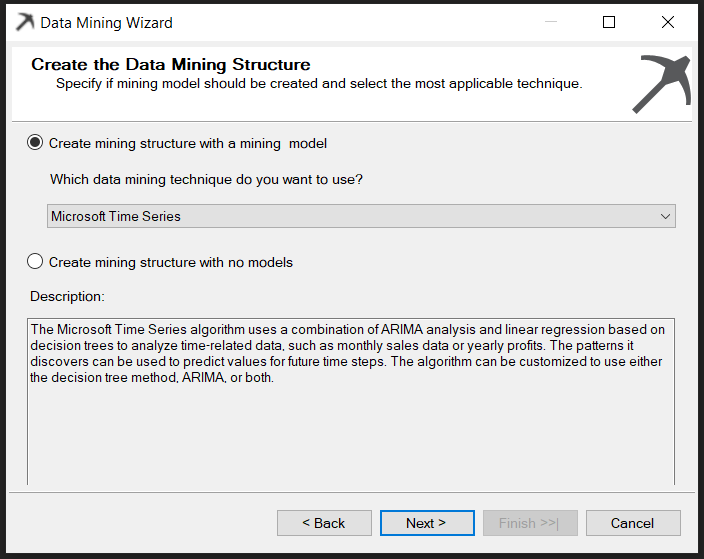
Kết quả cuối cùng:



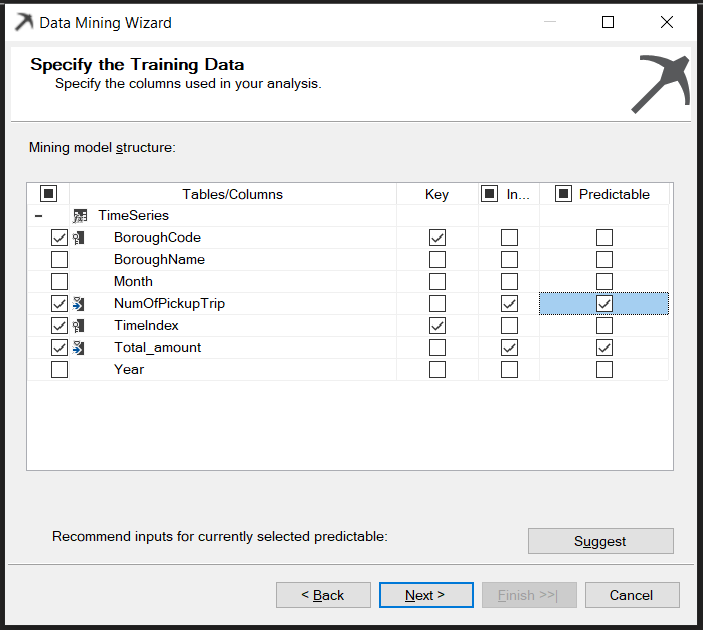
Như vậy bảng trên đã đủ điều kiện để mining bằng thuật toán TimeSeries.

**Tiến hành Mining:**

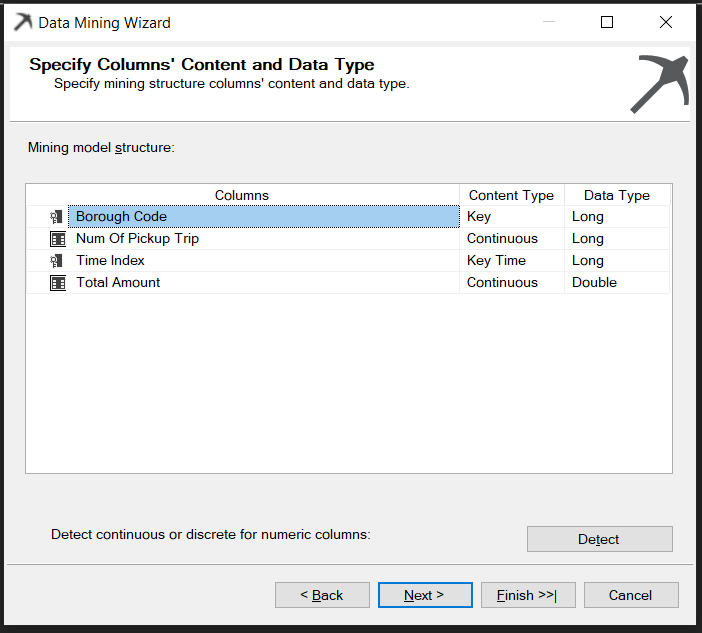
Tạo Mining Structures -> Microsoft Time Series



Cấu hình key, input, predictable

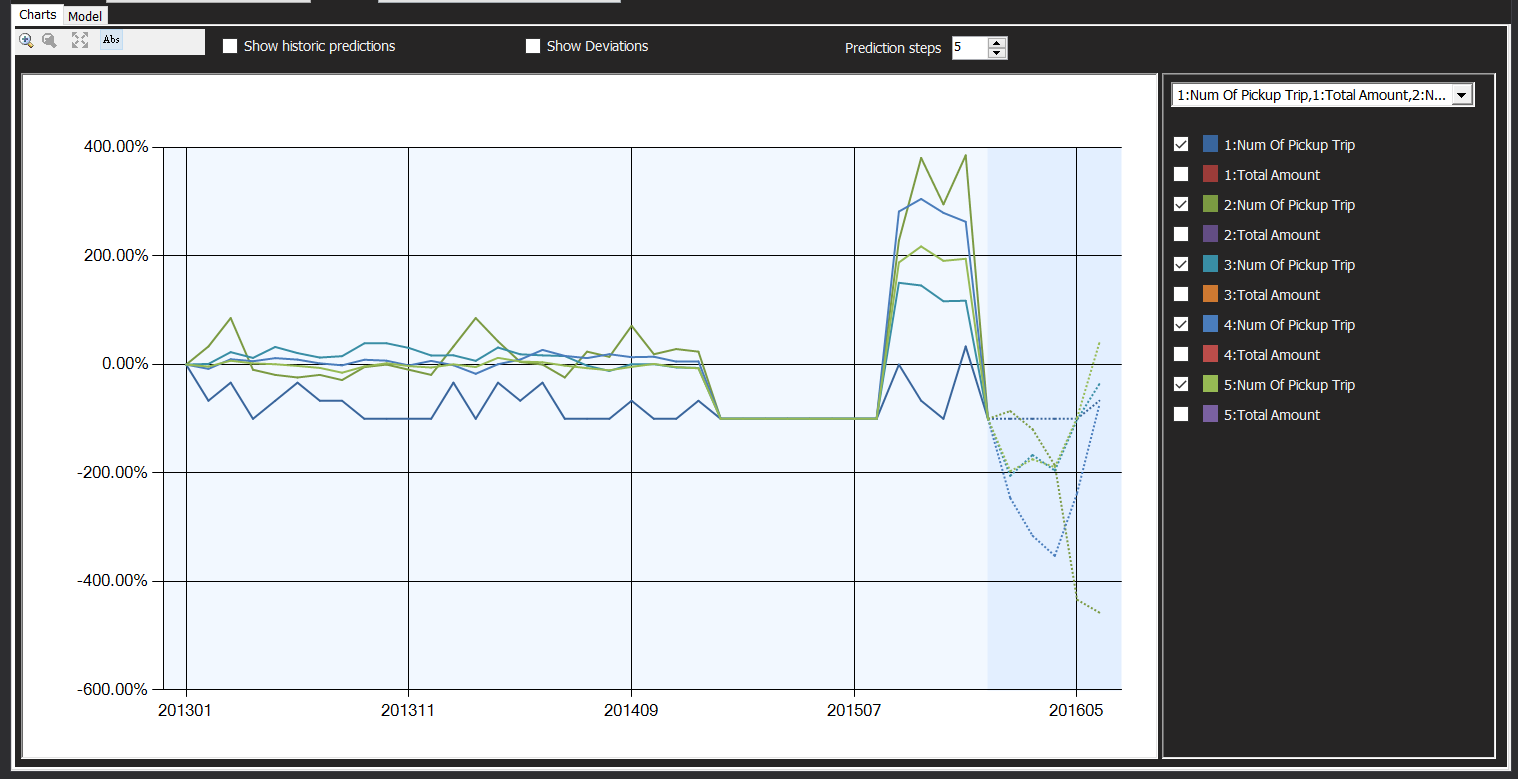


Cấu hình kiểu dữ liệu:

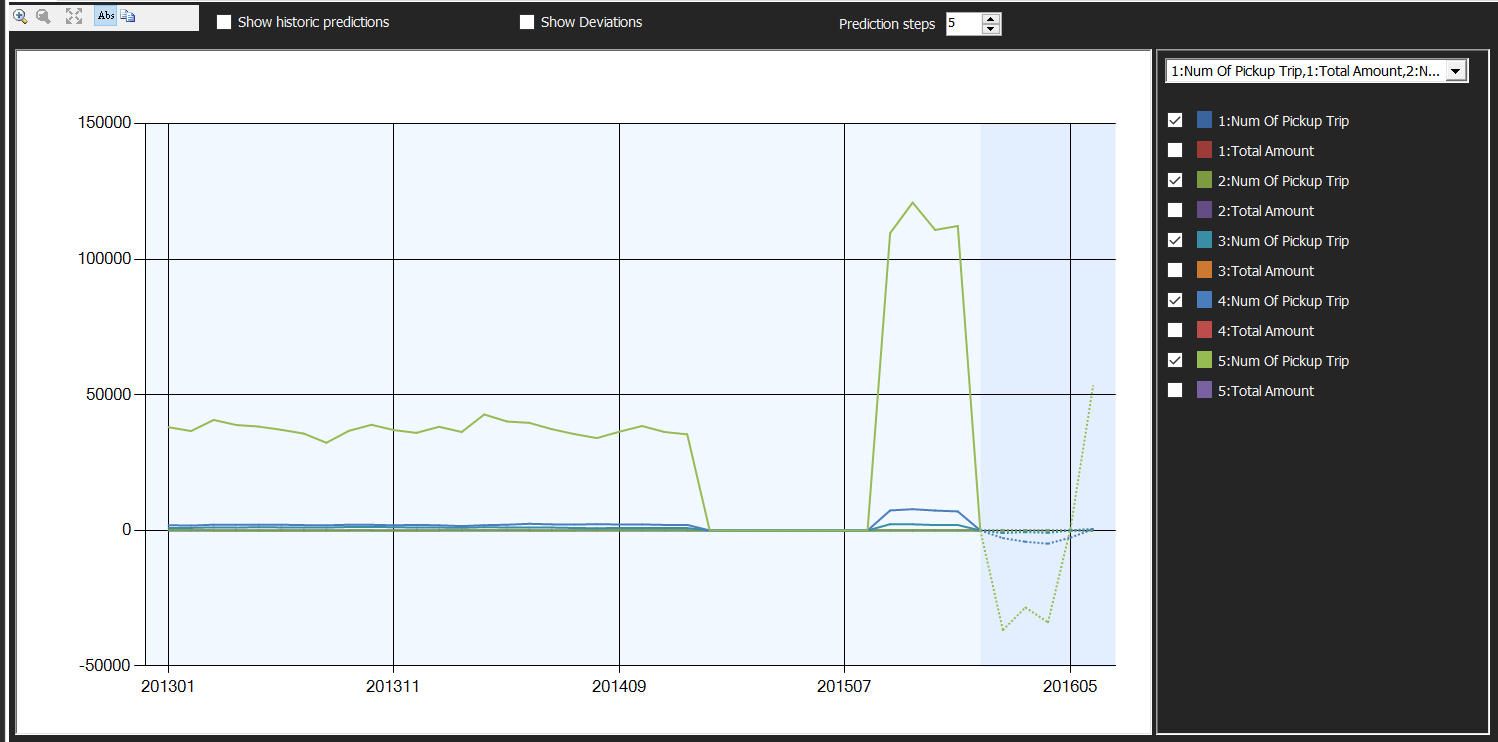


Kết quả:

Chênh lệch lượng đặt xe của các quận theo thời gian và dự đoán(nét đứt)



Số lượng đặt xe của các quận theo thời gian và dự đoán(nét đứt)



Chênh lệch doanh thu các quận thời gian và dự đoán(nét đứt)



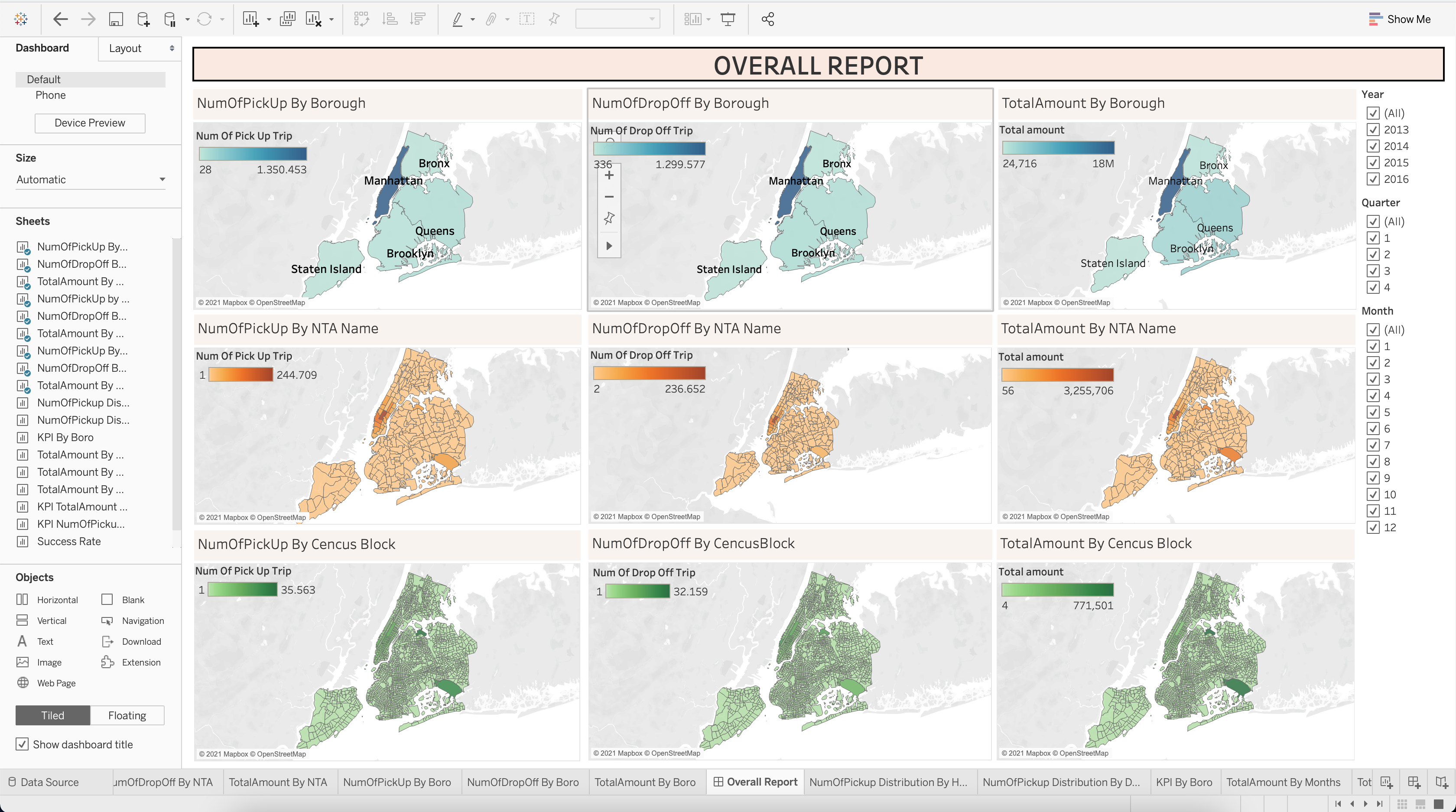
Doanh thu của các quận theo thời gian và dự đoán(nét đứt)



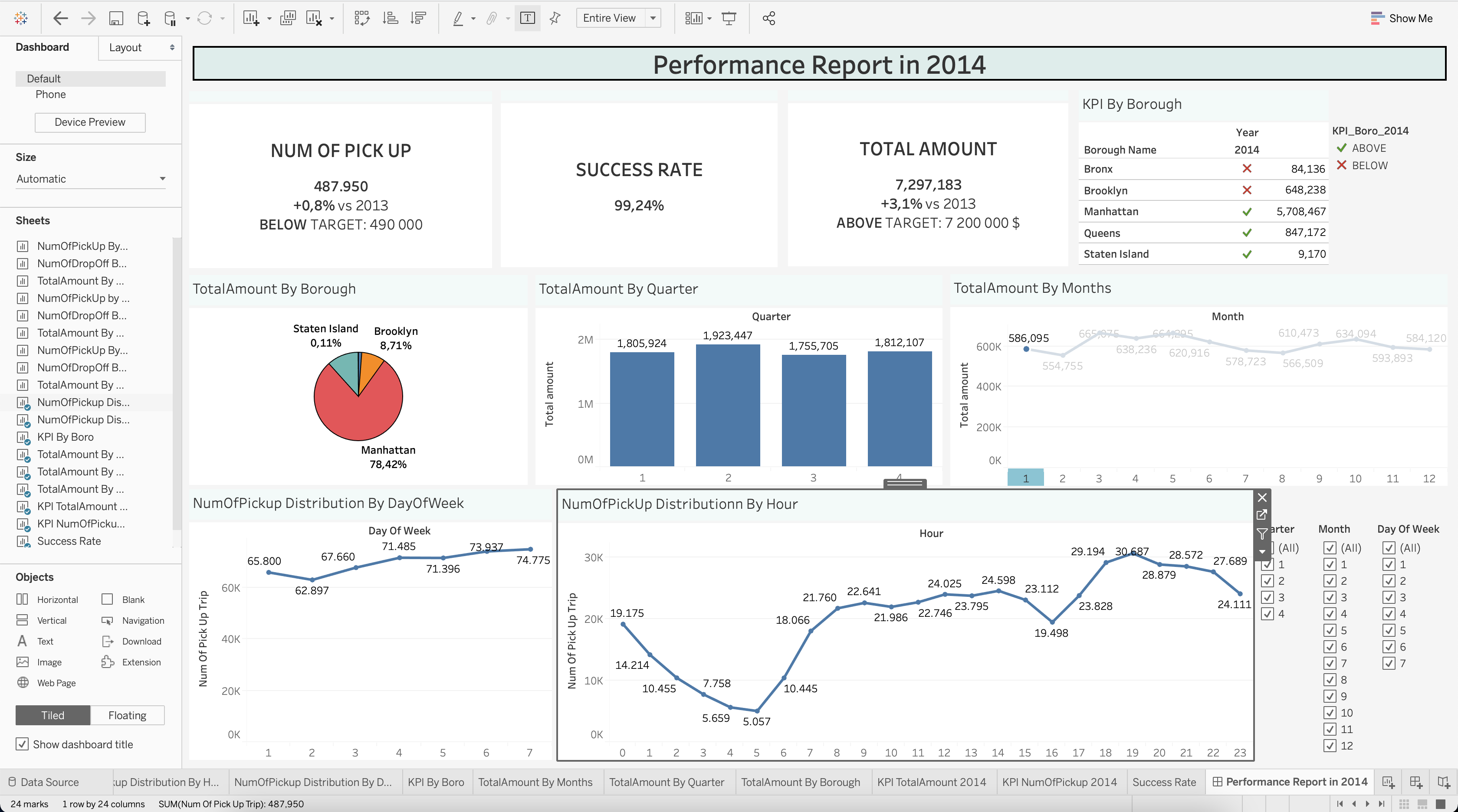
# TABLEAU REPORT DASHBOARD

Đây là Dashboard về lượng pickup/dropoff/totalAmount ở CencusBlock/NTA/Boro ở tất cả các năm.

Có thể filter theo year/quarter/month ở cột bên phải để nhìn dưới nhiều chiều thời gian.



Dưới đây là Report riêng cho năm 2014



Bao gồm:

* **NUM OF PICK UP:** Cho thấy tổng lượng đặt xe của năm 2014, so sánh với lượng đặt xe năm 2013, đạt được Target: 490 000 hay không.
* **SUCCESS RATE:** Cho biết % đặt xe thành công trong năm 2014
* **TOTAL AMOUNT:** Cho thấy được tổng doanh số của năm 2014, tăng giảm bao nhiêu % so với năm 2013, đạt được Target: 7 200 000USD hay không?
* **KPI By Borough:** Doanh số năm 2014 của từng quận, và cho biết có đạt được chỉ tiêu hay không
* **TotalAmount By Borough:** Cho biết % doanh số của 5 quận
* **TotalAmount By Quarter:** Cho biết doanh số qua từng quý
* **TotalAmount By Month:** Cho biết doanh số qua từng tháng biến động như thế nào
* **NumOfPickup Distribution By DayOfWeek:** Cho biết sự phân bổ lượng đặt xe của các ngày trong tuần trong năm 2014
* **NumOfPickup Distrubution By Hour**: Cho biết sự phân bổ lượng đặt xe của mỗi giờ trong một ngày

Ngoài ra có thể filter theo quý/tháng/ngàytrongtuần để nhìn dữ liệu một cách chi tiết hơn.