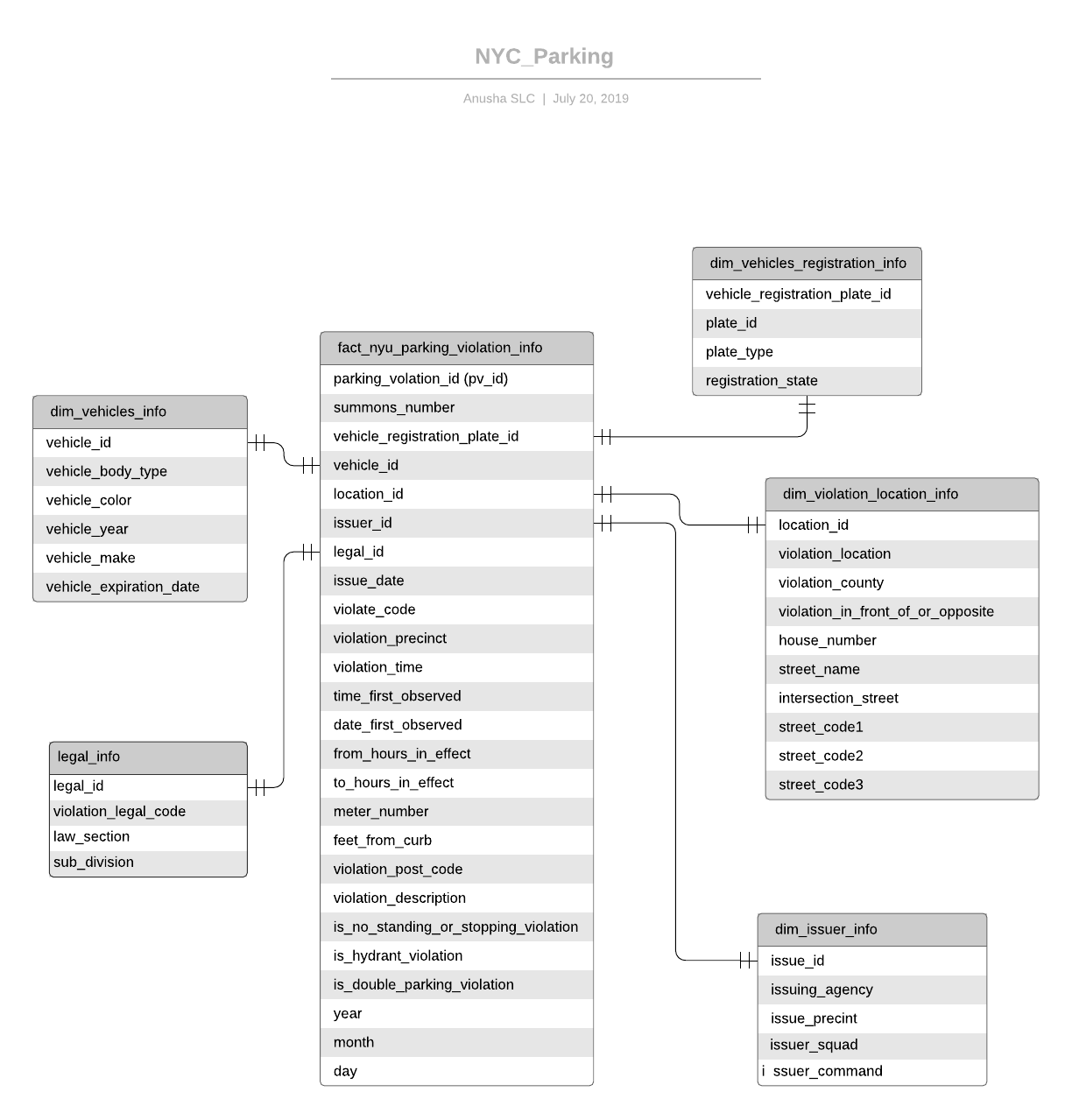
**Data Mart - NYC Parking Violation**

**Design**

Fact Table is parking\_violation\_ticket\_info :

* Fact table is partitioned based on state (NYC) and year/month/day
* Aggregate Columns:
  + Total violations in day
  + Total no stand/stop violations in day
  + Total hydrant violations in day
  + Total double parking violations in day
  + Max parking days
  + Min parking time
  + Max feet from curb
  + Min feet from curb

**IMPLEMENTED ER DIAGRAM (STAR SCHEMA)**

[****](https://www.lucidchart.com/documents/edit/eaefca9c-f449-4e01-be63-ab156c2684b2/0?callback=close&name=docs&callback_type=back&v=3639&s=612)

**Scenarios:**

* Find if the plate used is valid, what if fake plate is used?
* Find number of hydrants in the area, are they within the city limits?
* Accuracy of NTA device, do you need to replace the device?
* What’s causing more parking violations, is it because there’s no public parking nearby?
* Cause for more parking violations in the area, is it because new ice cream/ retail shop opened or airport nearby or event happening in the area?
* Can you make more money if you open a commercial parking garage in the area?
* Are the violations valid?
* Which agency issued violations more in a year?

**Batch Processing:**

The job runs daily to aggregate the data for the aggregation columns specified above

**ETL (Notebook) Progress:**

**Fact and Dimension tables ETL job:**

Created dimensions and facts and stored as parquet. Fact is stored based on partition column(year/month/day).

Due to limitations in data storage, only 2017 issue date year is processed

**Database:**

dm\_nyu\_parking

**Dimension Tables and Columns:**

dim\_vehicles\_registration\_info:

vehicle\_registration\_plate\_id:long

plate\_id:string

registration\_state:string

unregistered\_vehicle:integer

Plate\_type:string

dim\_issuer\_info:

issuer\_id:long

issuer\_precinct:integer

issuer\_code:integer

issuer\_command:string

Issuer\_squad:string

issuing\_agency:string

dim\_vehicles\_info:

vehicle\_id:long

vehicle\_body\_type:string

vehicle\_make:string

vehicle\_year:integer

Vehicle\_color:string

vehicle\_expiration\_date:integer

dim\_violation\_location\_info:

location\_id:long

violation\_location:integer

violation\_county:string

violation\_in\_front\_of\_or\_opposite:string

house\_number:string

street\_name:string

intersecting\_street:string

street\_code1:integer

street\_code2:integer

Street\_code3:integer

dim\_legal\_info:

legal\_id:long

law\_section:integer

sub\_division:string

Violation\_legal\_code:string

**Fact table columns:**

Fact\_nyu\_parking\_violation\_info:

Parking\_violation\_id:bigint

summons\_number:long

issue\_date:date

Violation\_code:integer

violation\_precinct:integer

violation\_time:string

time\_first\_observed:string

date\_first\_observed:integer

days\_parking\_in\_effect:string

from\_hours\_in\_effect:string

to\_hours\_in\_effect:string

unregistered\_vehicle:integer

meter\_number:string

feet\_from\_curb:integer

violation\_post\_code:string

violation\_description:string

no\_standing\_or\_stopping\_violation:string

hydrant\_violation:string

double\_parking\_violation:string

vehicle\_registration\_plate\_id:long

issuer\_id:long

location\_id:long

vehicle\_id:long

legal\_id:long

year:integer

month:integer

Day:integer

I’ve used a counter for dimension id but you can also use hashkey but counter, which is integer is faster than hashkey, disadvantage of counter, in my logic, is it’s not continuous increment.

**Code:**

[**https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/5041140880827814/4014305827219043/2252038084290521/latest.html**](https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/5041140880827814/4014305827219043/2252038084290521/latest.html)

**Agg ETL job:**

agg\_nyu\_parking\_violation\_info:

year:integer

month:integer

day:integer

violation\_time:string

location\_id:long

issuer\_id:long

vehicle\_registration\_plate\_id:long

violation\_description:string

count\_summons\_number:long

count\_no\_stand\_number:long

count\_hydrant\_violation\_number:long

count\_double\_parking\_violation\_number:long

max\_days\_parking\_in\_effect\_number:string

min\_days\_parking\_in\_effect\_number:string

max\_feet\_from\_curb\_number:integer

Min\_feet\_from\_curb\_number:integer

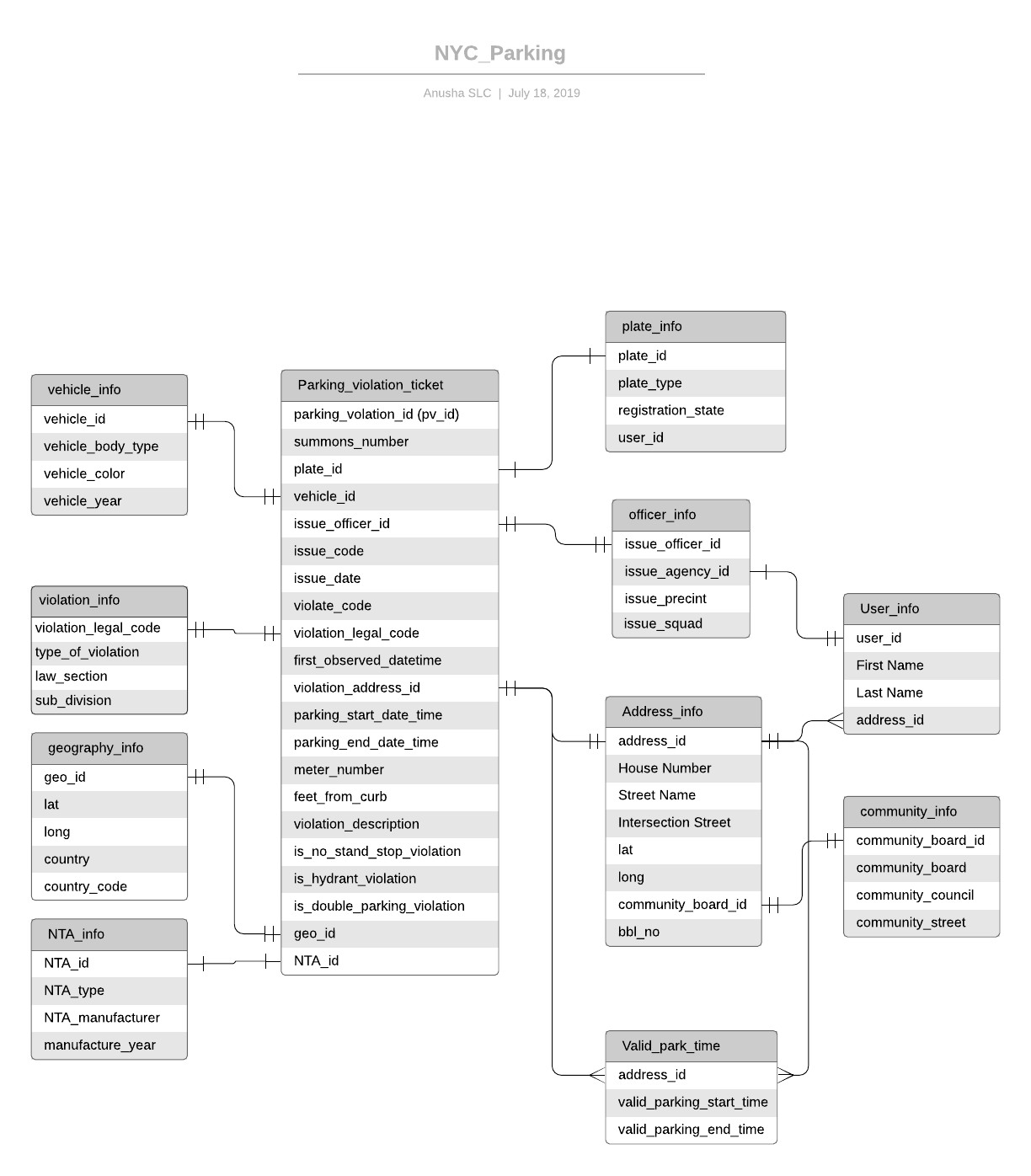
**Aggregrate columns:**

1. Total violations in day
2. Total no stand/stop violations in day
3. Total hydrant violations in day
4. Total double parking violations in day
5. Max parking days
6. Min parking time
7. Max feet from curb
8. Min feet from curb

**Code:**

[**https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/5041140880827814/3333566614432119/2252038084290521/latest.html**](https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/5041140880827814/3333566614432119/2252038084290521/latest.html)

**IDEAL ER DIAGRAM** (SNOWFLAKE SCHEMA)

[](https://www.lucidchart.com/documents/edit/eaefca9c-f449-4e01-be63-ab156c2684b2/0?callback=close&name=docs&callback_type=back&v=2421&s=612)

**Query Results**

**With Fact**

[**https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/5041140880827814/4014305827219070/2252038084290521/latest.html**](https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/5041140880827814/4014305827219070/2252038084290521/latest.html)

**With Agg**

[**https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/5041140880827814/2634234997705643/2252038084290521/latest.html**](https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/5041140880827814/2634234997705643/2252038084290521/latest.html)