311 Complaint: Street Condition

&

Motor Vehicle Collisions Crashes

Narrative Description:

The first dataset that was selected contains 311 service requests from 2010 to present in New York City. According to the 311 Service Requests data, in 2017, street condition was the 5th most common complaint issue with a total of 93,270 complaints for the year alone. Therefore, this specific complaint type was chosen to explore whether vehicle accidents are more likely to occur depending on road conditions.

Street condition complaints could be key indicators in predicting the likelihood of collisions and essential in preventing future crashes. As a result, a second dataset has been selected that contains the details on motor vehicle collision events in New York City. By merging these two datasets, we hope to find correlations between complaints of road conditions, which included potholes, failed street repairs, and more, and vehicle collisions.

For instance, with the information that we have, we hope to easily identify which locations should often receive inspections to assess the road conditions due to the higher volume of complaints. As well as what time of the year would streets have hazardous road conditions? For instance, during the fourth quarter of the year where there are more snow storms, streets would be more damaged than the rest of the year. By having access to this information, it would be easier for us to prepare and improve the road conditions and thus, reduce vehicle accidents drastically.

Data Sources

1. 311 Service Requests
2. Motor Vehicle Collisions Crashes

Extracted Sample Data:

1. 311 Service Requests
2. Motor Vehicle Collisions

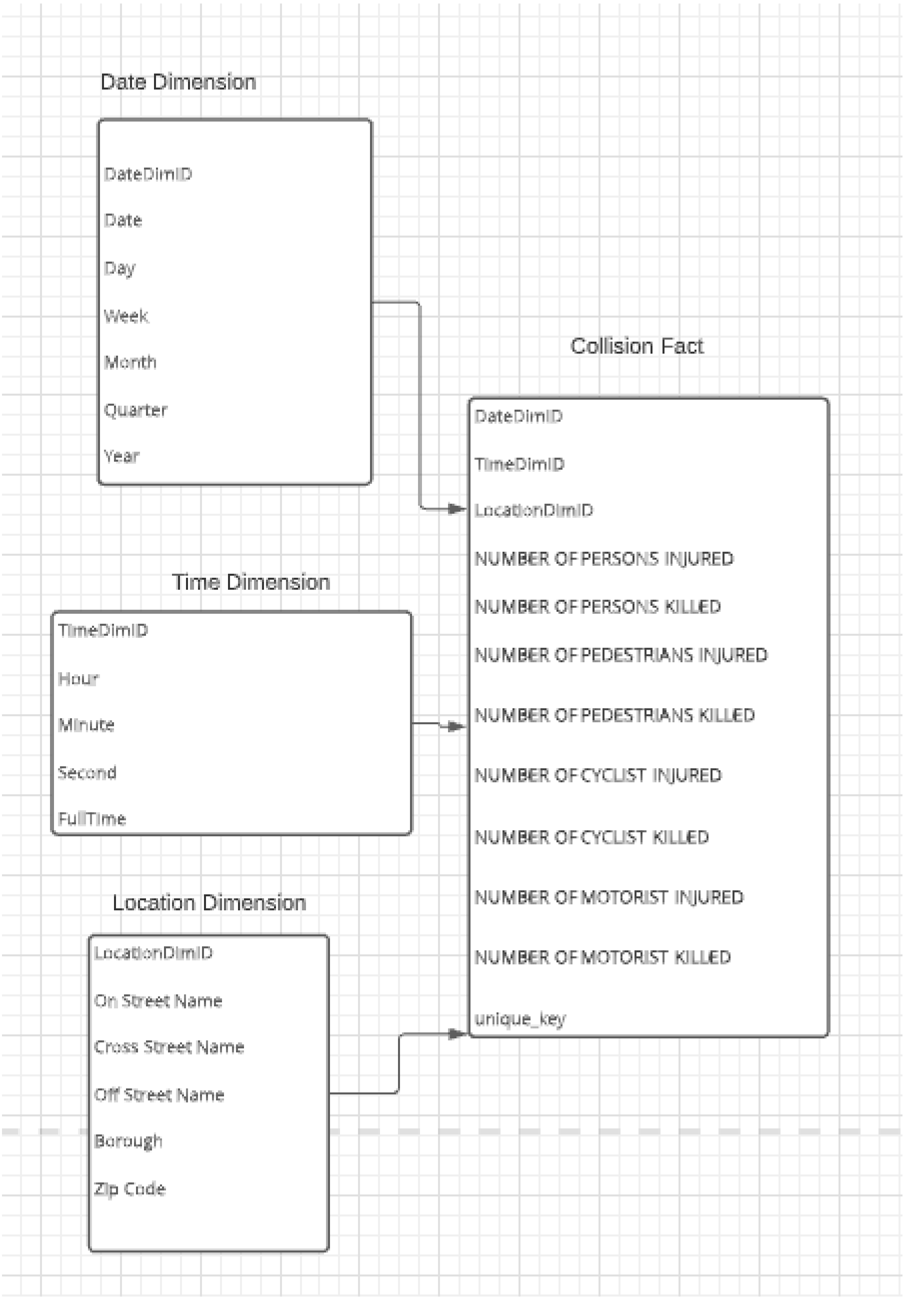
KPIs

1. Street Condition Complaints and Collisions Count by Zip Code
2. Number of Persons Killed By Borough
3. Number of Persons Injured by Borough
4. Number of Deaths by Road Condition Description
5. Occurrences of Road Condition Complaints Based on Collisions

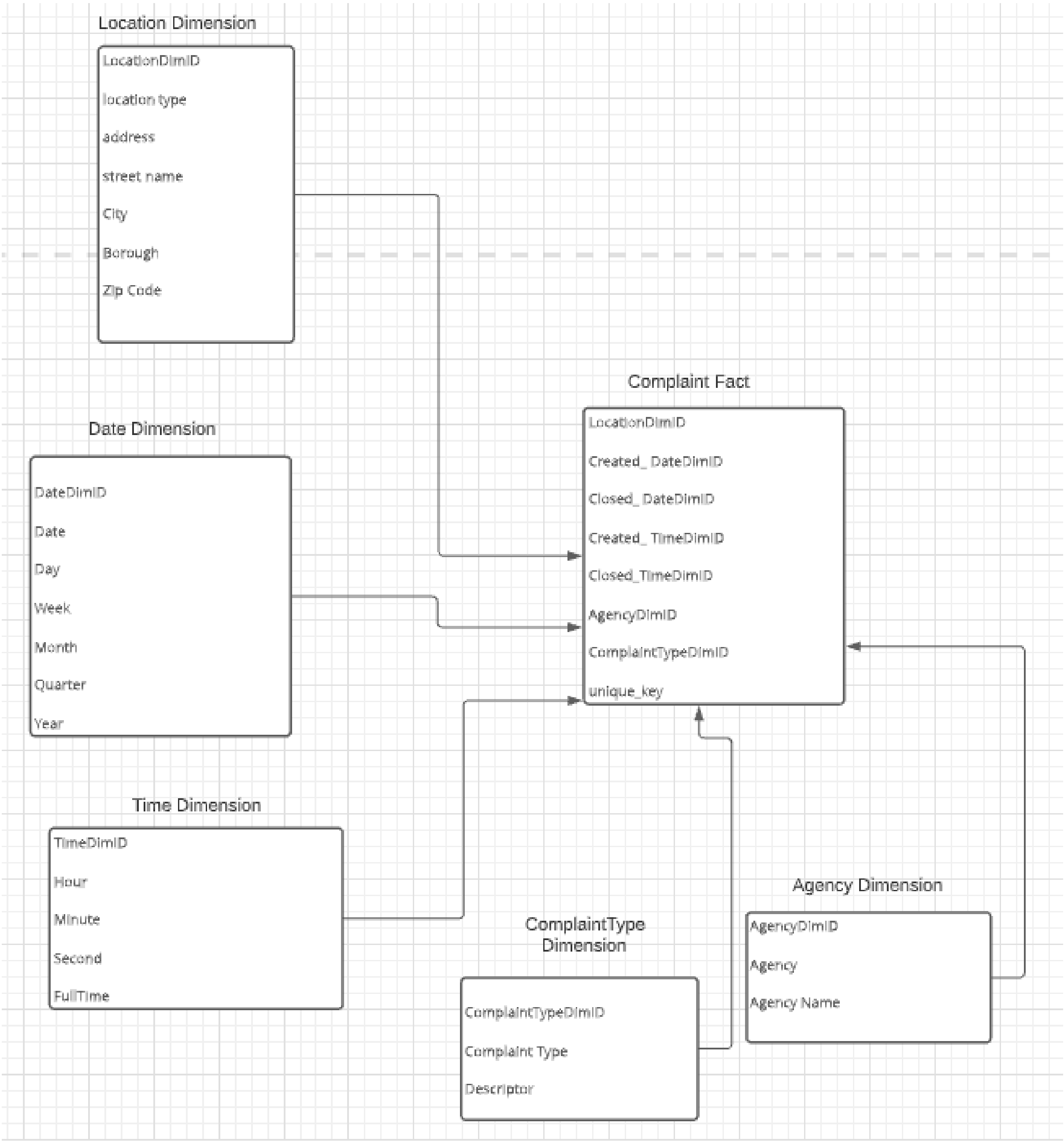
Dimensional Model: Link to LucidChart Dimensional Model

Some KPIs involve frequency monitoring based on time. Therefore, transaction grain has been selected for every record in the fact table for each of the transactions. The only attribute that would most likely change is location; however, infrequently. Thus, type 3 slowly changing dimensions has been selected.

Collision Fact

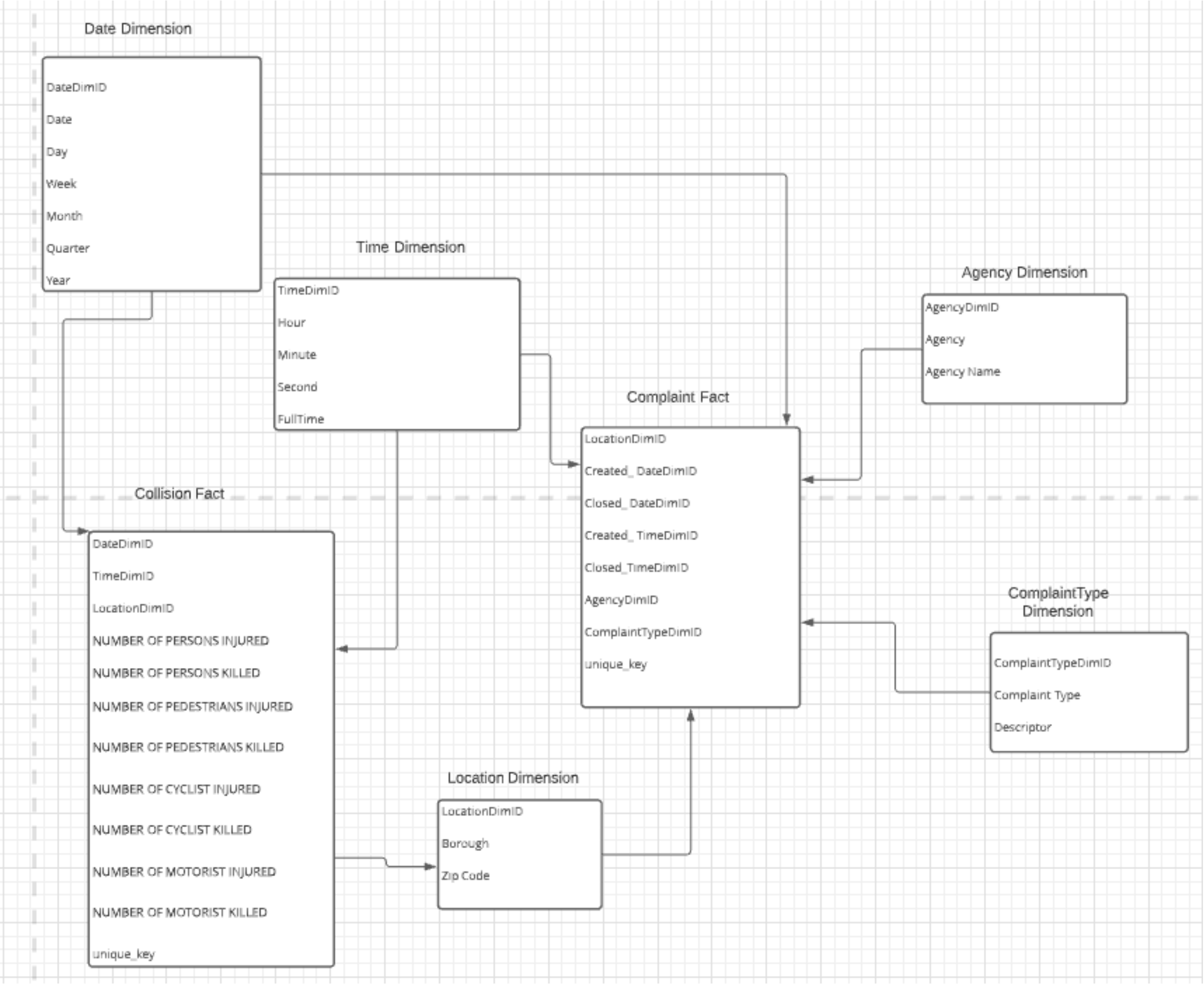


Complaint Fact



Final Dimensional Schema (Integrated Model)

The Collision and Complaint fact tables share Date, Time and Location dimensions. The Complaint fact table also contains Agency and ComplaintType dimensions.



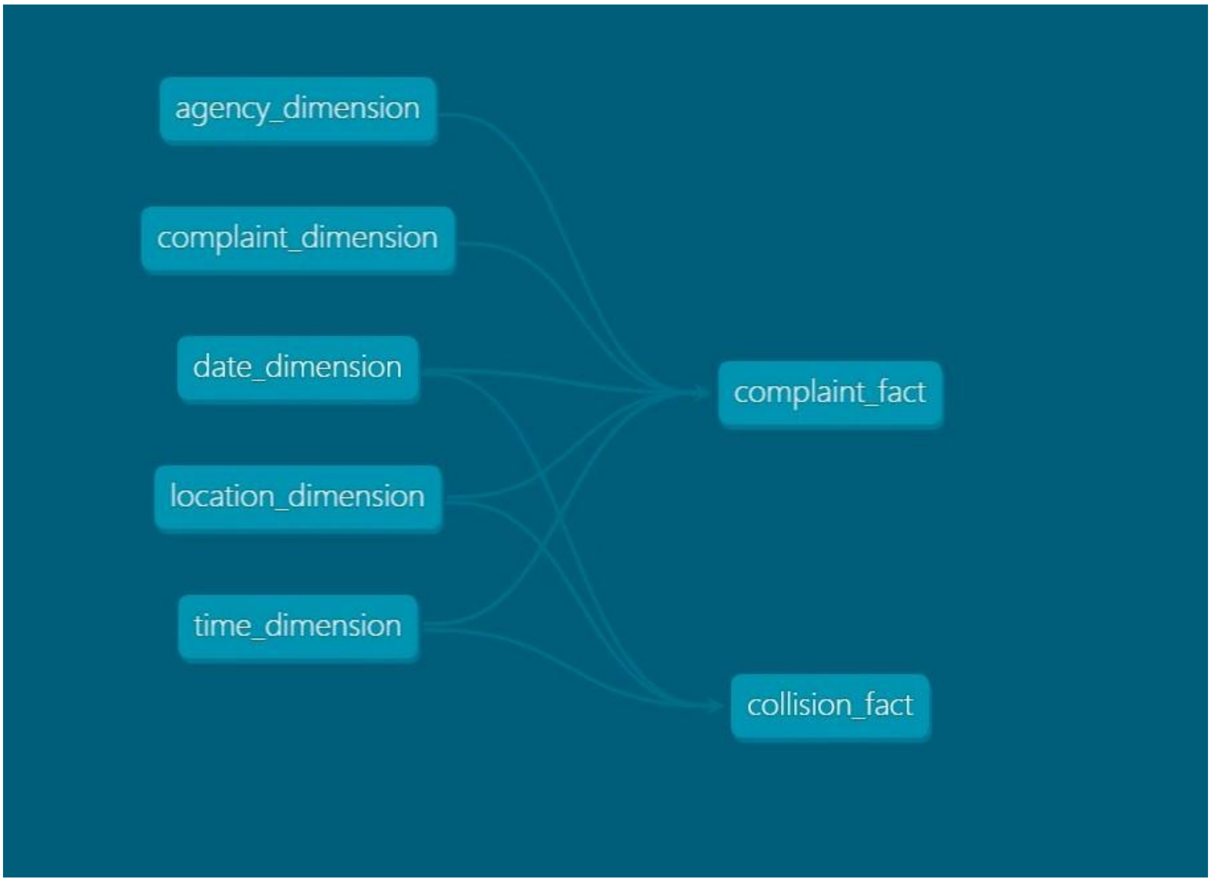
Data Profiling: Data Profiling

A Pandas module was used for profiling. This was a quick exploratory analysis step we took before proceeding. These multiple visualisations saved a lot of time and gave us an understanding of the distribution of each variable.

ETL Programming For Dimensions and Fact Tables:

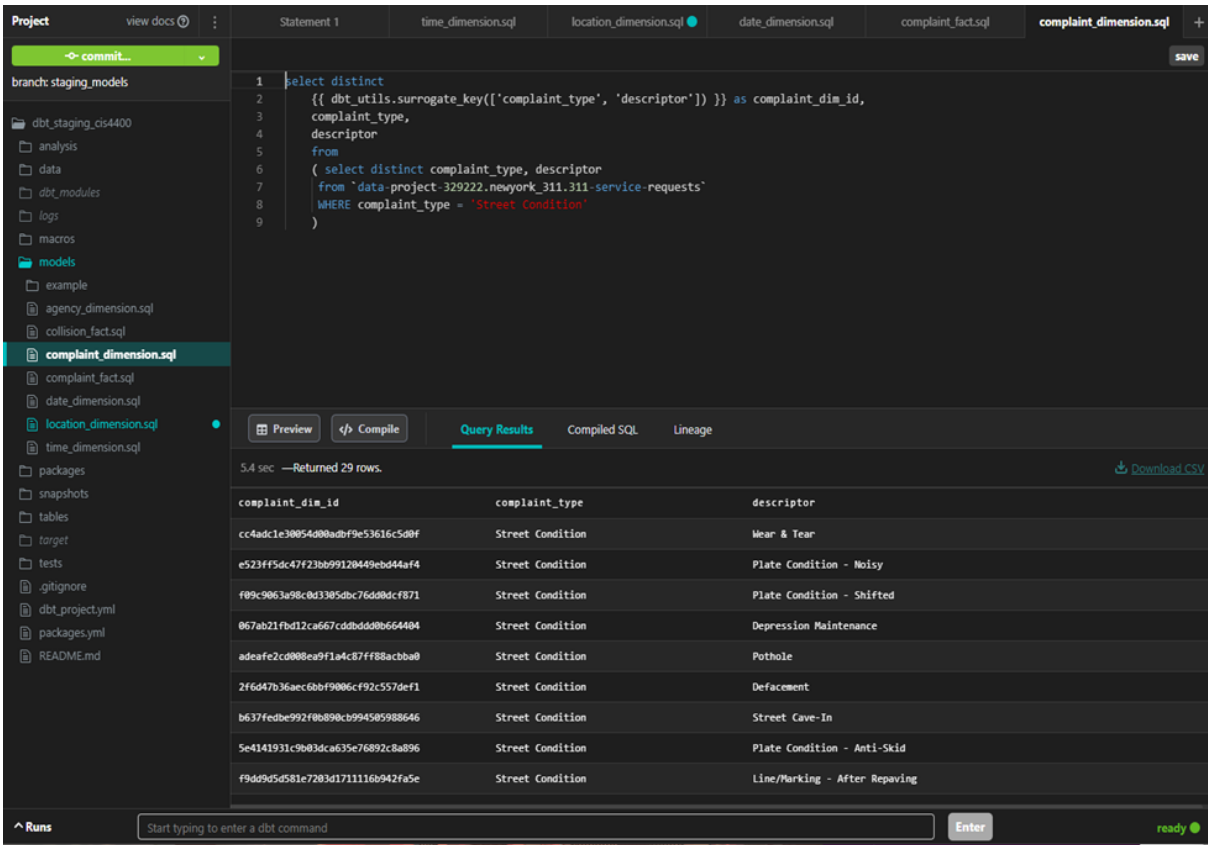
Lineage Graph

With the ETL tool DBT, we created our dimension models and fact tables. The diagram shows the relationship between each dimension to the fact tables.



Complaint Type Dimension

A surrogate key is created and the complaint type and descriptor columns are created. It will also display these columns in the dimension. Our “where” clause only takes records for the specific complaint type, “Street Condition”.



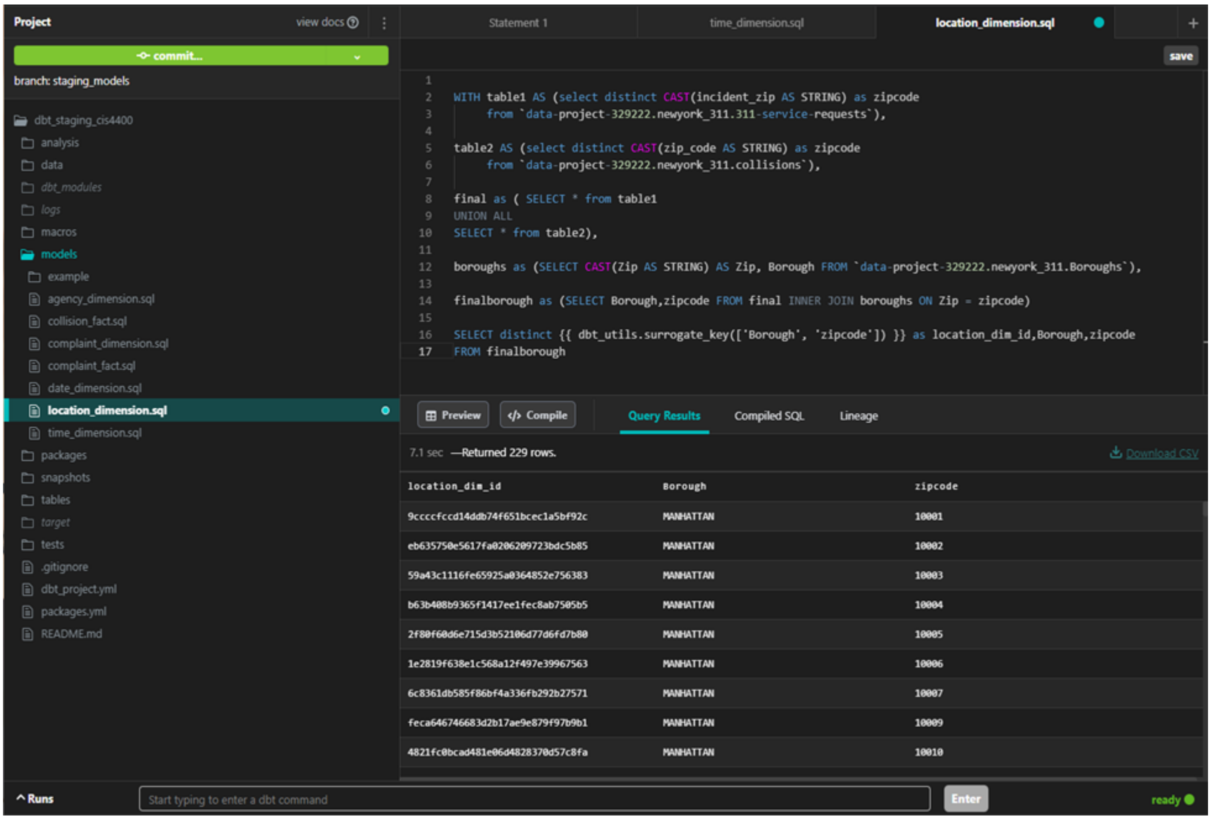
```sql
select distinct
    {{ dbt_utils.surrogate_key(['complaint_type', 'descriptor']) }} as complaint_dim_id,
    complaint_type,
    descriptor
    from
    ( select distinct complaint_type, descriptor
     from `data-project-329222.newyork_311.311-service-requests`
     WHERE complaint_type = 'Street Condition'
    )
```

5.4 sec —Returned 29 rows.

complaint_dim_id	complaint_type	descriptor
cc4adc1e30054d00adbf9e53616c5d0f	Street Condition	Wear & Tear
e523ff5dc47f23bb99120449ebd44af4	Street Condition	Plate Condition - Noisy
f09c9063a98c0d3305dbc76dd0dcf871	Street Condition	Plate Condition - Shifted
067ab21fbd12ca667cddbddd0b664404	Street Condition	Depression Maintenance
adeafe2cd008ea9f1a4c87ff88acbba0	Street Condition	Pothole
2f6d47b36aec6bbf9006cf92c557def1	Street Condition	Defacement
b637fedbe992f0b890cb994505988646	Street Condition	Street Cave-In
5e4141931c9b03dca635e76892c8a896	Street Condition	Plate Condition - Anti-Skid
f9dd9d5d581e7203d1711116b942fa5e	Street Condition	Line/Marking - After Repaving

Location Dimension

The datasets are integrated by zip code and borough. After unioning the separate location tables for the datasets, the location dimension was created with a surrogate key.



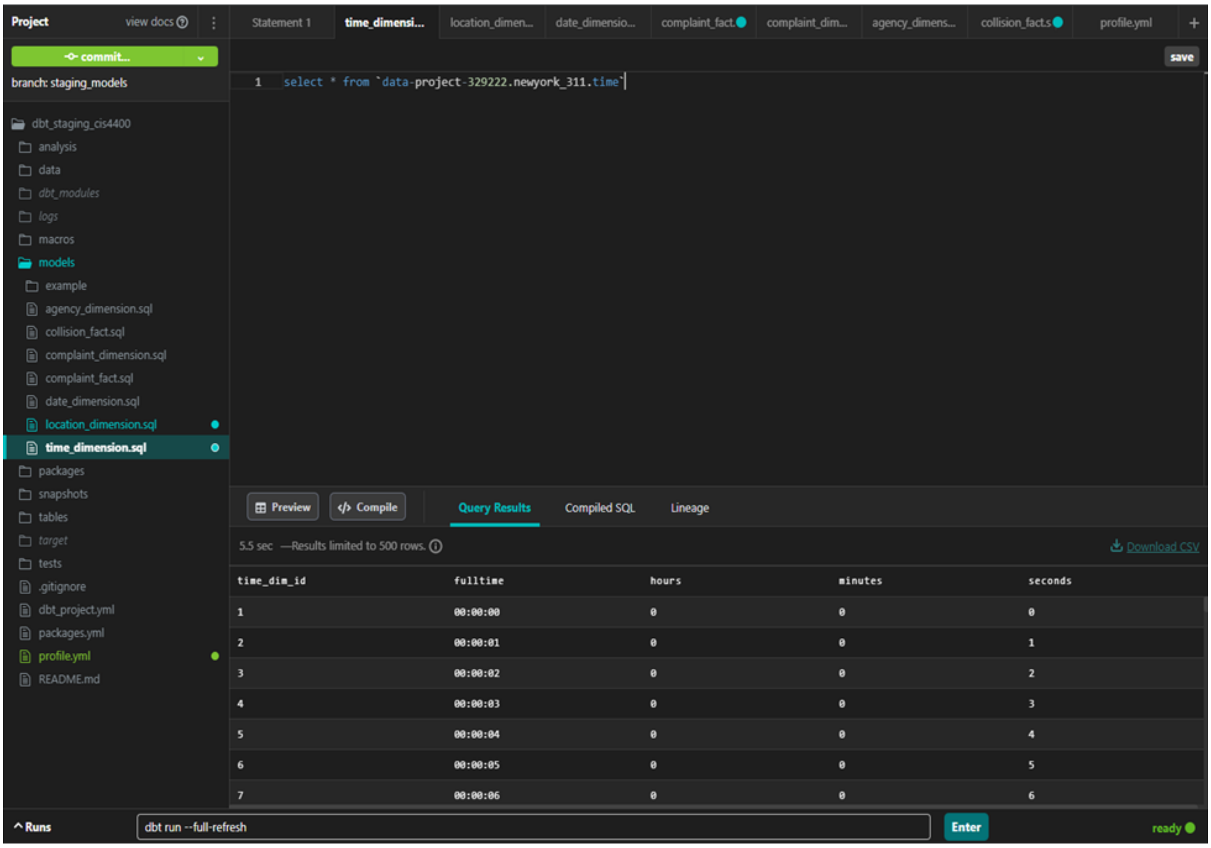
```

WITH table1 AS (select distinct CAST(incident_zip AS STRING) as zipcode
    from `data-project-329222.newyork_311.311-service-requests`),

table2 AS (select distinct CAST(zip_code AS STRING) as zipcode
    from `data-project-329222.newyork_311.collisions`),

final as ( SELECT * from table1
UNION ALL
SELECT * from table2),

boroughs as (SELECT CAST(Zip AS STRING) AS Zip, Borough FROM `data-project-329222.newyork_311.Boroughs`),

finalborough as (SELECT Borough,zipcode FROM final INNER JOIN boroughs ON Zip = zipcode)

SELECT distinct {{ dbt_utils.surrogate_key(['Borough', 'zipcode']) }} as location_dim_id,Borough,zipcode
FROM finalborough
```

7.1 sec —Returned 229 rows.

location_dim_id	Borough	zipcode
9ccccfccd14ddb74f651bcec1a5bf92c	MANHATTAN	10001
eb635750e5617fa0206209723bdc5b85	MANHATTAN	10002
59a43c1116fe65925a0364852e756383	MANHATTAN	10003
b63b408b9365f1417ee1fec8ab7505b5	MANHATTAN	10004
2f80f60d6e715d3b52106d77d6fd7b80	MANHATTAN	10005
1e2819f638e1c568a12f497e39967563	MANHATTAN	10006
6c8361db585f86bf4a336fb292b27571	MANHATTAN	10007
feca646746683d2b17ae9e879f97b9b1	MANHATTAN	10009
4821fc0bcad481e06d4828370d57c8fa	MANHATTAN	10010

Time Dimension

A CSV file was integrated that contains time_dim_id, fulltime, hours, minutes, and seconds. It was uploaded to BigQuery and selected all records from this dataset to create the time dimension.



```
select * from `data-project-329222.newyork_311.time`
```

5.5 sec —Results limited to 500 rows.

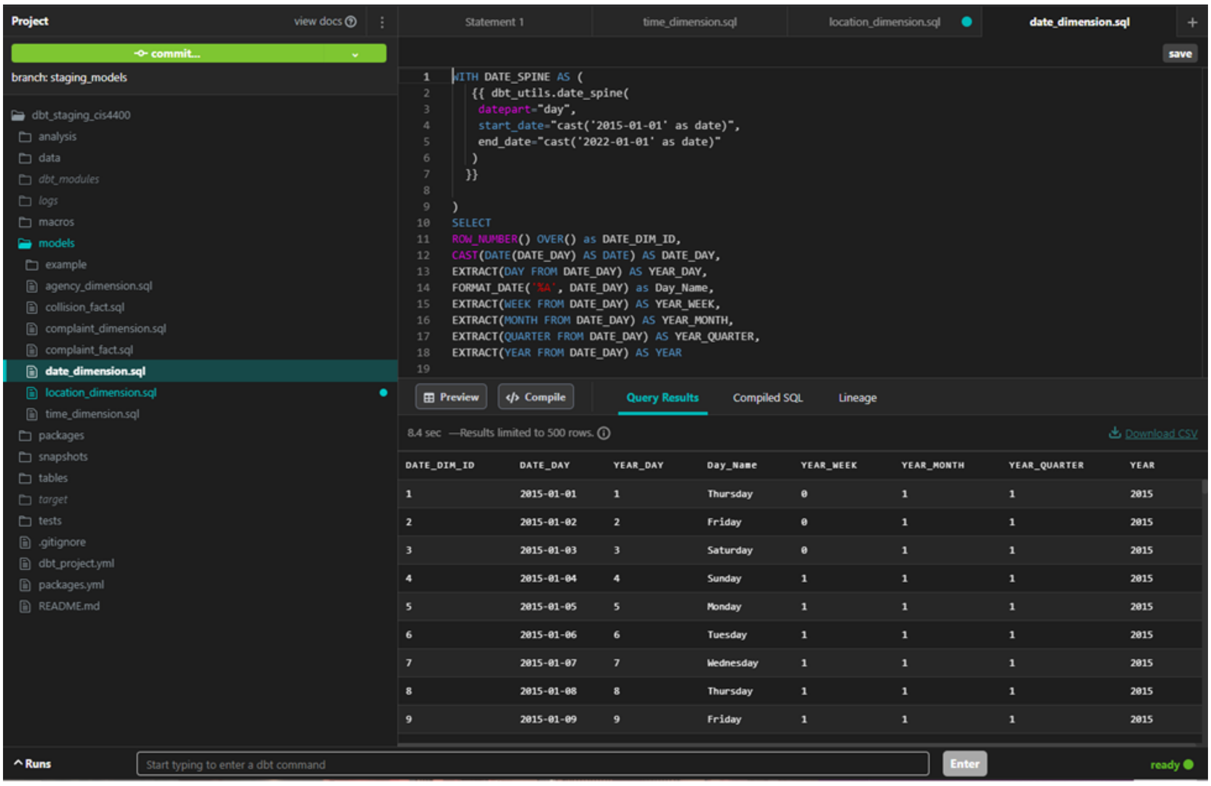
time_dim_id	fulltime	hours	minutes	seconds
1	00:00:00	0	0	0
2	00:00:01	0	0	1
3	00:00:02	0	0	2
4	00:00:03	0	0	3
5	00:00:04	0	0	4
6	00:00:05	0	0	5
7	00:00:06	0	0	6

```
dbt run --full-refresh
```

Date Dimension

The dbt utils package was installed to construct a date dimension with the dimension ID, full date, the number sequence of day (out of 365 days) in the year, the day of the week, the number sequence of week (out of 52 weeks) the day falls in in the year, the month of the year, the quarter in the year, and the year.

Code Source: gitlab-data/analytics/date



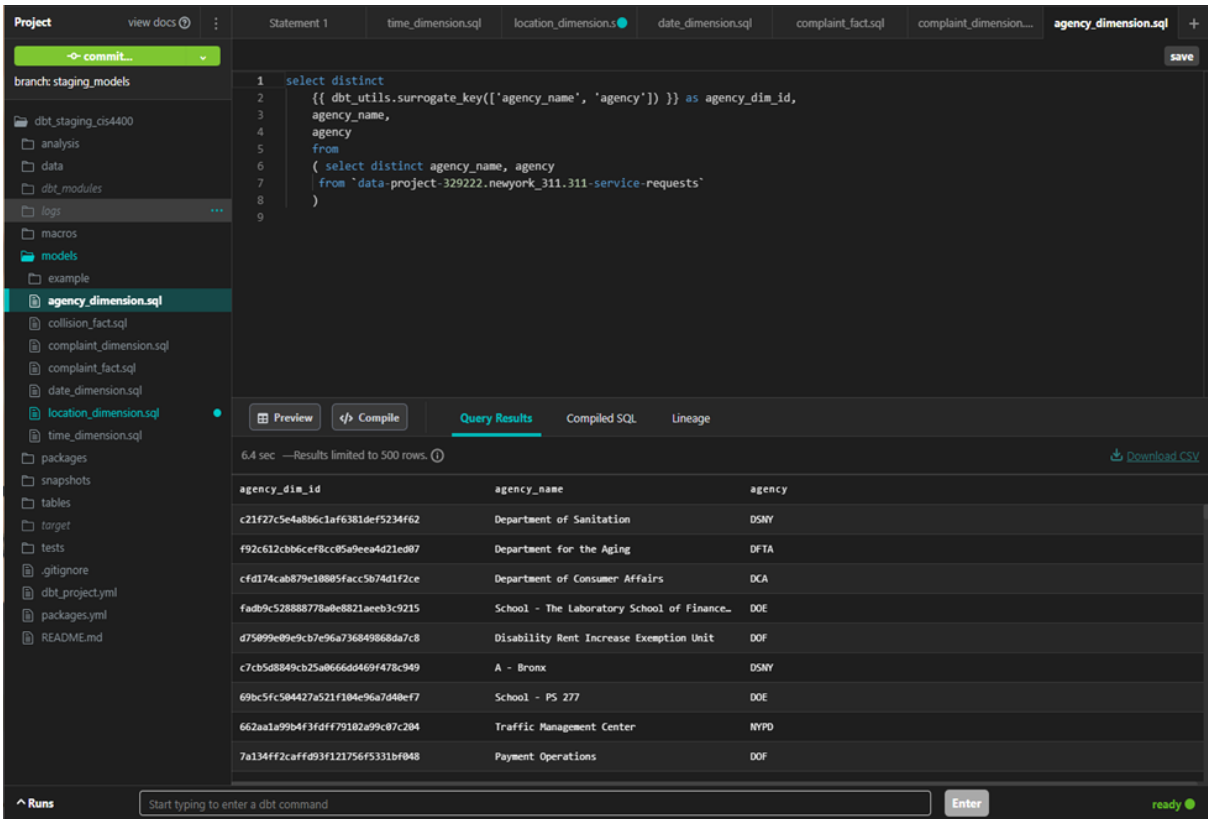
```sql
WITH DATE_SPINE AS (
   {{ dbt_utils.date_spine(
    datepart="day",
    start_date="cast('2015-01-01' as date)",
    end_date="cast('2022-01-01' as date)"
   )
  }}

)
SELECT
ROW_NUMBER() OVER() as DATE_DIM_ID,
CAST(DATE(DATE_DAY) AS DATE) AS DATE_DAY,
EXTRACT(DAY FROM DATE_DAY) AS YEAR_DAY,
FORMAT_DATE('%A', DATE_DAY) as Day_Name,
EXTRACT(WEEK FROM DATE_DAY) AS YEAR_WEEK,
EXTRACT(MONTH FROM DATE_DAY) AS YEAR_MONTH,
EXTRACT(QUARTER FROM DATE_DAY) AS YEAR_QUARTER,
EXTRACT(YEAR FROM DATE_DAY) AS YEAR

```

8.4 sec —Results limited to 500 rows.

DATE_DIM_ID	DATE_DAY	YEAR_DAY	Day_Name	YEAR_WEEK	YEAR_MONTH	YEAR_QUARTER	YEAR
1	2015-01-01	1	Thursday	0	1	1	2015
2	2015-01-02	2	Friday	0	1	1	2015
3	2015-01-03	3	Saturday	0	1	1	2015
4	2015-01-04	4	Sunday	1	1	1	2015
5	2015-01-05	5	Monday	1	1	1	2015
6	2015-01-06	6	Tuesday	1	1	1	2015
7	2015-01-07	7	Wednesday	1	1	1	2015
8	2015-01-08	8	Thursday	1	1	1	2015
9	2015-01-09	9	Friday	1	1	1	2015

Agency Dimension

Agency name and agency as the surrogate key from the 311 Service Requests dataset. It will also display these columns with the original values in the dimension.



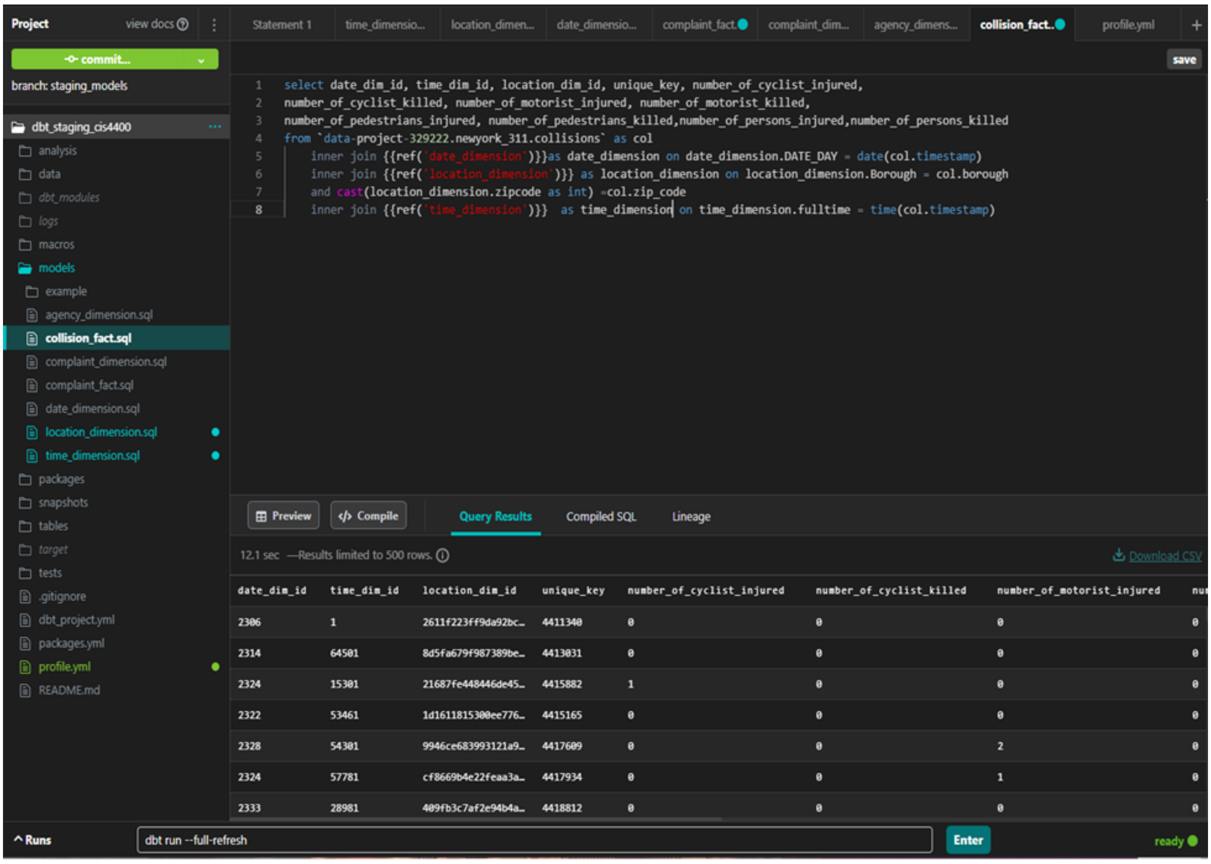
```sql
select distinct
    {{ dbt_utils.surrogate_key(['agency_name', 'agency']) }} as agency_dim_id,
    agency_name,
    agency
    from
    ( select distinct agency_name, agency
     from `data-project-329222.newyork_311.311-service-requests`
    )
```

6.4 sec —Results limited to 500 rows.

agency_dim_id	agency_name	agency
c21f27c5e4a8b6c1af6381def5234f62	Department of Sanitation	DSNY
f92c612cbb6cef8cc05a9eea4d21ed07	Department for the Aging	DFTA
cfd174cab879e10805facc5b74d1f2ce	Department of Consumer Affairs	DCA
fadb9c528888778a0e8821aeeb3c9215	School - The Laboratory School of Finance…	DOE
d75099e09e9cb7e96a736849868da7c8	Disability Rent Increase Exemption Unit	DOF
c7cb5d8849cb25a0666dd469f478c949	A - Bronx	DSNY
69bc5fc504427a521f104e96a7d40ef7	School - PS 277	DOE
662aa1a99b4f3fdff79102a99c07c204	Traffic Management Center	NYPD
7a134ff2caffd93f121756f5331bf048	Payment Operations	DOF

Collision Fact

We have selected the dimensions ID, the degenerated dimension, and the columns from the Collision dataset that will add valuable information to create the KPIs. The datasets are joined with the related dimensions (Date, Location, and Time) on the shared columns in order to display the table.



```sql
select date_dim_id, time_dim_id, location_dim_id, unique_key, number_of_cyclist_injured,
number_of_cyclist_killed, number_of_motorist_injured, number_of_motorist_killed,
number_of_pedestrians_injured, number_of_pedestrians_killed,number_of_persons_injured,number_of_persons_killed
from `data-project-329222.newyork_311.collisions` as col
    inner join {{ref('date_dimension')}}as date_dimension on date_dimension.DATE_DAY = date(col.timestamp)
    inner join {{ref('location_dimension')}} as location_dimension on location_dimension.Borough = col.borough
    and cast(location_dimension.zipcode as int) =col.zip_code
    inner join {{ref('time_dimension')}}  as time_dimension on time_dimension.fulltime = time(col.timestamp)
```

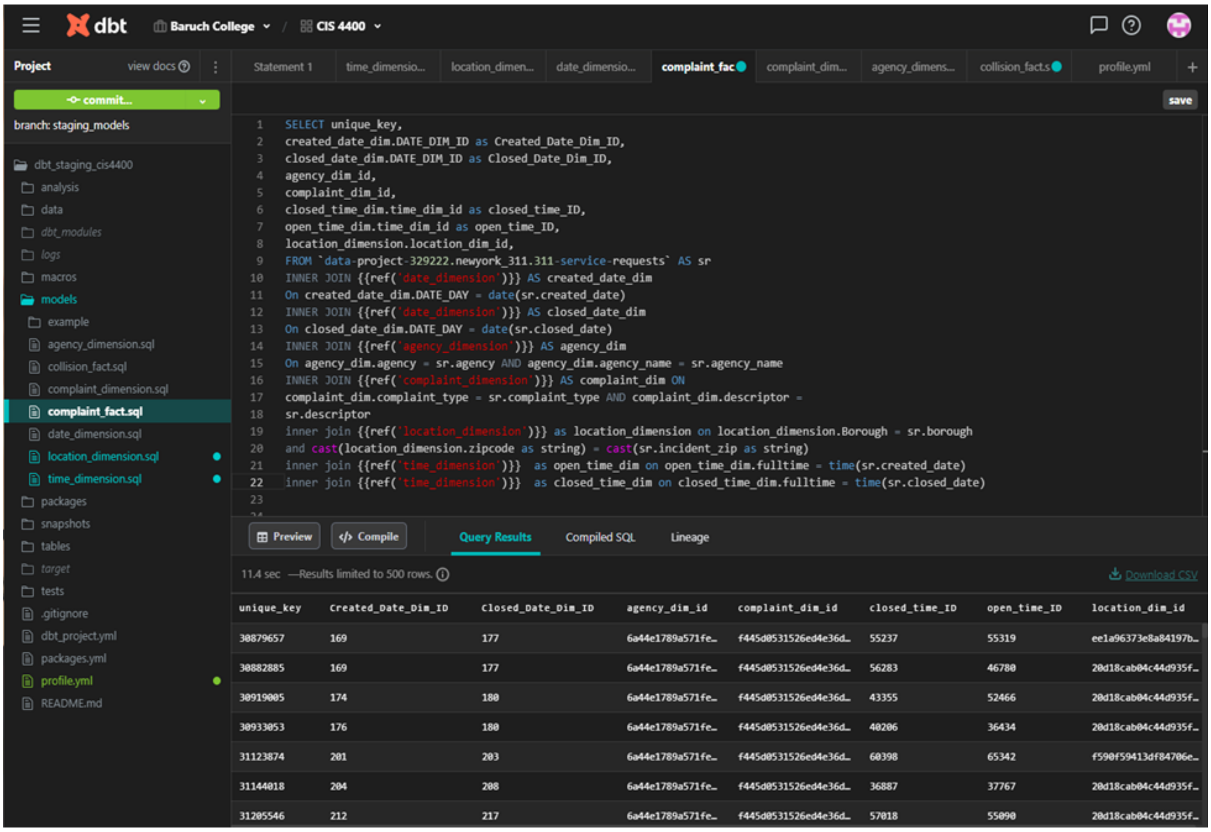
12.1 sec —Results limited to 500 rows.

date_dim_id	time_dim_id	location_dim_id	unique_key	number_of_cyclist_injured	number_of_cyclist_killed	number_of_motorist_injured	nu
2306	1	2611f223ff9da92bc...	4411340	0	0	0	0
2314	64501	8d5fa679f987389be...	4413031	0	0	0	0
2324	15301	21687fe448446de45...	4415882	1	0	0	0
2322	53461	1d1611815300ee776...	4415165	0	0	0	0
2328	54301	9946ce683993121a9...	4417609	0	0	2	0
2324	57781	cf8669b4e22feaa3a...	4417934	0	0	1	0
2333	28981	409fb3c7af2e94b4a...	4418812	0	0	0	0

```
dbt run --full-refresh
```

Complaint Fact

We have selected the dimension IDs and the degenerated dimension that are critical for this fact table. The dataset are joined with the related dimensions (Date, Location, Agency, Complaint_Type, and Time) on the shared columns in order to display the table.



```sql
SELECT unique_key,
created_date_dim.DATE_DIM_ID as Created_Date_Dim_ID,
closed_date_dim.DATE_DIM_ID as Closed_Date_Dim_ID,
agency_dim_id,
complaint_dim_id,
closed_time_dim.time_dim_id as closed_time_ID,
open_time_dim.time_dim_id as open_time_ID,
location_dimension.location_dim_id,
FROM `data-project-329222.newyork_311.311-service-requests` AS sr
INNER JOIN {{ref('date_dimension')}} AS created_date_dim
On created_date_dim.DATE_DAY = date(sr.created_date)
INNER JOIN {{ref('date_dimension')}} AS closed_date_dim
On closed_date_dim.DATE_DAY = date(sr.closed_date)
INNER JOIN {{ref('agency_dimension')}} AS agency_dim
On agency_dim.agency = sr.agency AND agency_dim.agency_name = sr.agency_name
INNER JOIN {{ref('complaint_dimension')}} AS complaint_dim ON
complaint_dim.complaint_type = sr.complaint_type AND complaint_dim.descriptor =
sr.descriptor
inner join {{ref('location_dimension')}} as location_dimension on location_dimension.Borough = sr.borough
and cast(location_dimension.zipcode as string) = cast(sr.incident_zip as string)
inner join {{ref('time_dimension')}}  as open_time_dim on open_time_dim.fulltime = time(sr.created_date)
inner join {{ref('time_dimension')}}  as closed_time_dim on closed_time_dim.fulltime = time(sr.closed_date)
```

11.4 sec —Results limited to 500 rows.

unique_key	Created_Date_Dim_ID	Closed_Date_Dim_ID	agency_dim_id	complaint_dim_id	closed_time_ID	open_time_ID	location_dim_id
30879657	169	177	6a44e1789a571fe...	f445d0531526ed4e36d...	55237	55319	ee1a96373e8a84197b...
30882885	169	177	6a44e1789a571fe...	f445d0531526ed4e36d...	56283	46780	20d18cab04c44d935f...
30919005	174	180	6a44e1789a571fe...	f445d0531526ed4e36d...	43355	52466	20d18cab04c44d935f...
30933053	176	180	6a44e1789a571fe...	f445d0531526ed4e36d...	40206	36434	20d18cab04c44d935f...
31123874	201	203	6a44e1789a571fe...	f445d0531526ed4e36d...	60398	65342	f590f59413df84706e...
31144018	204	208	6a44e1789a571fe...	f445d0531526ed4e36d...	36887	37767	20d18cab04c44d935f...
31205546	212	217	6a44e1789a571fe...	f445d0531526ed4e36d...	57018	55090	20d18cab04c44d935f...

Visualisations of KPIs: Presentation: Real Time Insights

- Data Lake: Google BigQuery
- Data Warehouse: Google BigQuery
- ETL Tools: DBT
- BI Application: Tableau
- Programming Language: SQL & Python
- Diagramming Application: LucidChart