Discovering the best places to live in NYC by analyzing 311 NYC Noise Complaint data and NYC Crime data

Kevin Linitz – kevin.linitz@baruchmail.cuny.edu

Thy Bui – thy.bui@baruchmail.cuny.edu

Raul Rachitoff – raul.rachitoffmiranda@baruchmail.cuny.edu

Arunima Dwivedi – arunima.dwivedi@baruchmail.cuny.edu

Group 5

CIS 9440 – Data Warehousing and Analytics

December 16, 2022

Introduction and Objectives

We are professionals that are looking to discover neighborhoods with preferably low levels of noise and high levels of safety. We are looking to eliminate the least fit area to live in NYC.

NYC is known to have high crime rates and lots of noise nuisance. Having said that, with elevated level of data skills, we are determined to optimize our choice by building a descriptive model that shows areas with the most crimes and noise to avoid.

Additionally, we want to analyze the correlation between noise and crimes and whether high noise complaints in an area leads to higher crime rates.

We will use two datasets, 311 NYC Noise Complaint data, and NYC Arrest data from the NYC Open-Source Dataset. The two datasets include data over the span from 2019-2022 which are updated by quarter, time granularity is day, and geo-granularity is geo-code.

KPIs:

1. Noise complaints (number of complaints) by month
2. Crime rate (number of arrests) by month
3. Noise complaint resolved rate (resolved complaints/total complaints)
4. Number of arrests by borough
5. Noise complaints by borough and Noise Complaints / Borough Population
6. Highest and Lowest Number of Arrests by Precinct
7. Percentage of arrests by age group, sex, and race

Datasets:

- 311 NYC Noise Complaint Data: https://data.cityofnewyork.us/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9
- NYC Crime historic Data: https://data.cityofnewyork.us/Public-Safety/NYPD-Arrests-Data-Historic-/8h9b-rp9u
- NYC Crime Year to Date Data: https://data.cityofnewyork.us/Public-Safety/NYPD-Arrest-Data-Year-to-Date-/uip8-fykc

Business Process:

- **Business Process / Event: Analyze NYC Complaint Service**
- **Business Process / Event: Analyze NYPD Arrests**

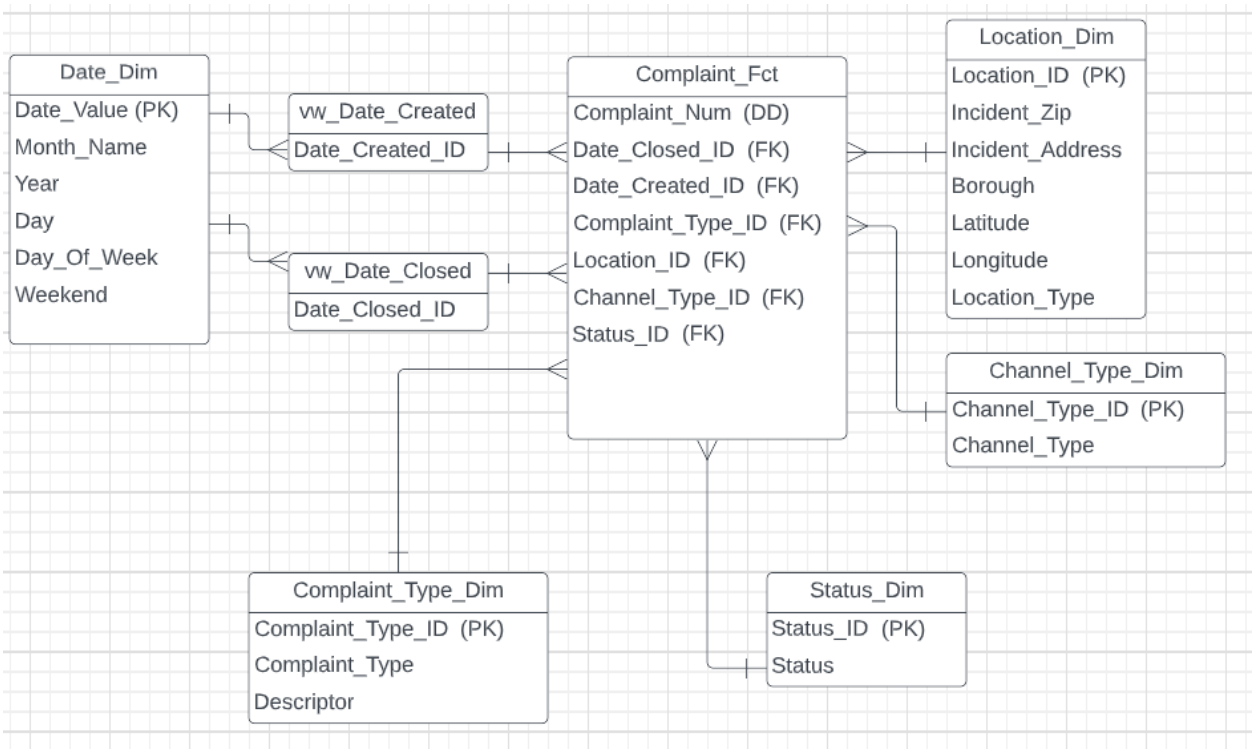
EDW Bus Matrix

	Complaints	Arrests	Date	Location	Channel Type	Complaint Type	Status	Offense Type	Age	Sex	Race
NYC 311 Complaint Service	X		X	X	X	X	X				
NYPD Arrests		X	X	X				X	X	X	X

Dimensional Modeling

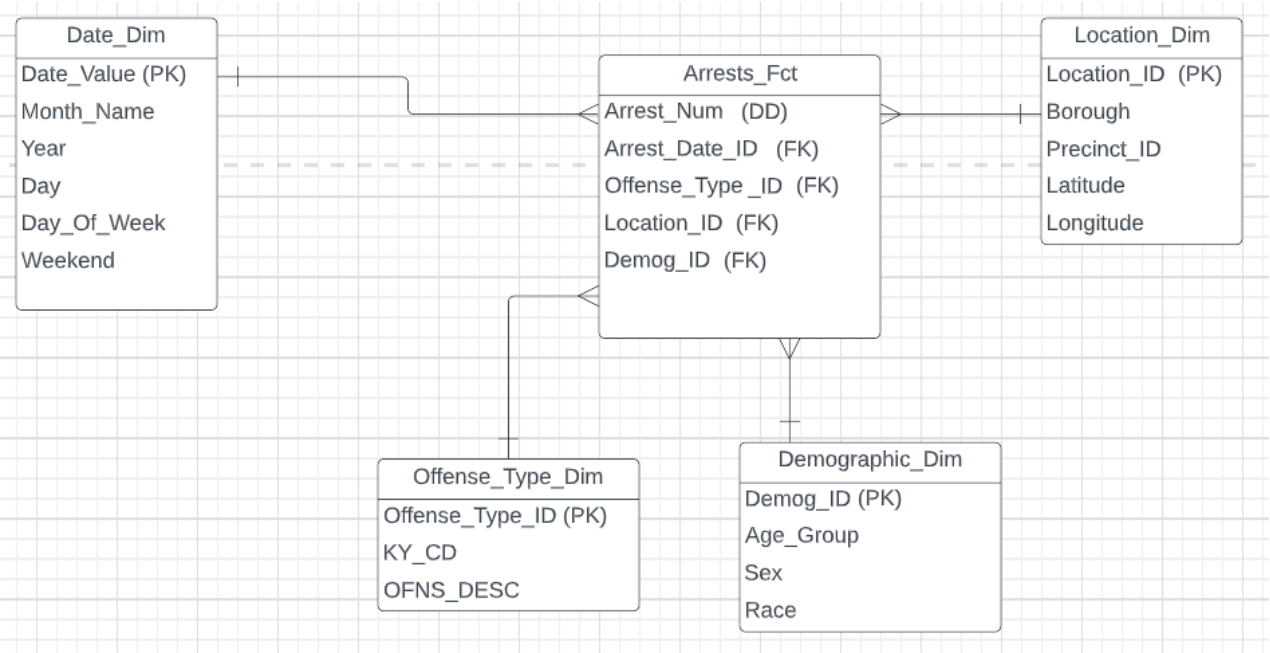
I. **Business Process / Event: Analyze NYC Complaint Service**

- **Grain**: one row for each Complaint Num



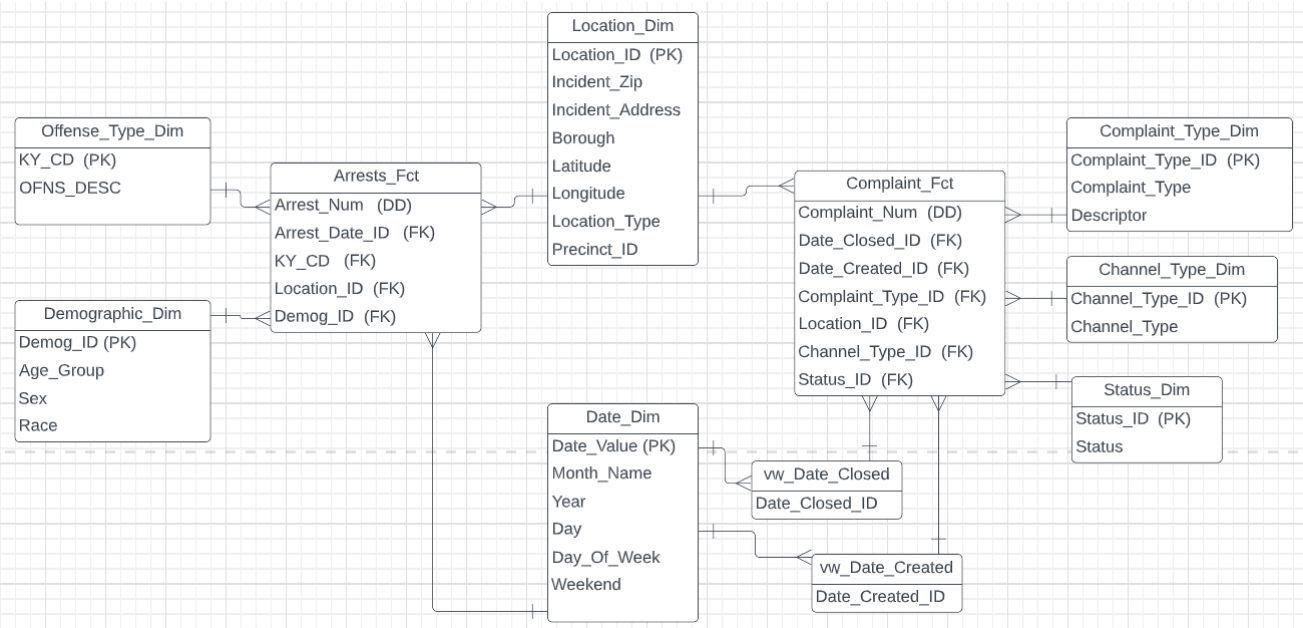
II. Business Process / Analyze Event: NYPD Arrests

Grain: one row for each Arrest



III. Integrated Datawarehouse Model

Grain: one row for each Arrest and Complaint



ETL Tools and Target DBMS Selected

Our team has selected **dbt** as our ETL tool and **Google BigQuery** as our Target DBMS for our project. We all have experience utilizing these tools and will build off the skills we have gained throughout this course. These tools will assist in transforming the thousands of rows of data from the 311 NYC Noise Complaint and NYC Crime datasets to help us get one step closer in discovering if there is correlation between noise complaints and crime rates.

For visualizing our KPIs we selected Tableau. Tableau is a powerful tool to easily illustrate our various KPI results. We also selected this tool because all members of our group have experience with Tableau since we all used this tool in the Data Visualization course at Baruch.

Streaming data from NYC Open Source to Google Cloud Platform using Socrata API:

We used python to stream data from NYC Open Source to GCP. This process allows us to quickly get historical data to GCP (5 minutes) and stream upcoming data.

Reference:

- DW Project-ExtractingNoiseData (file submitted with this PDF)
- DW project-ExtractingCrimeData (file submitted with this PDF)

Data Profiling

For data profiling we took a random sample of 20k records for each Noise & Crime dataset and generated the report through Pandas Profiling.

For data profiling and report after cleaning: DataProfiling-CleaningReport-Noise-Crime.html (file submitted with this PDF)

For data profiling results:

- pandas_profiling-NoiseData.html (file submitted with this PDF)
- pandas_profiling-CrimeData.html (file submitted with this PDF)

Cleaning report after data cleaning:

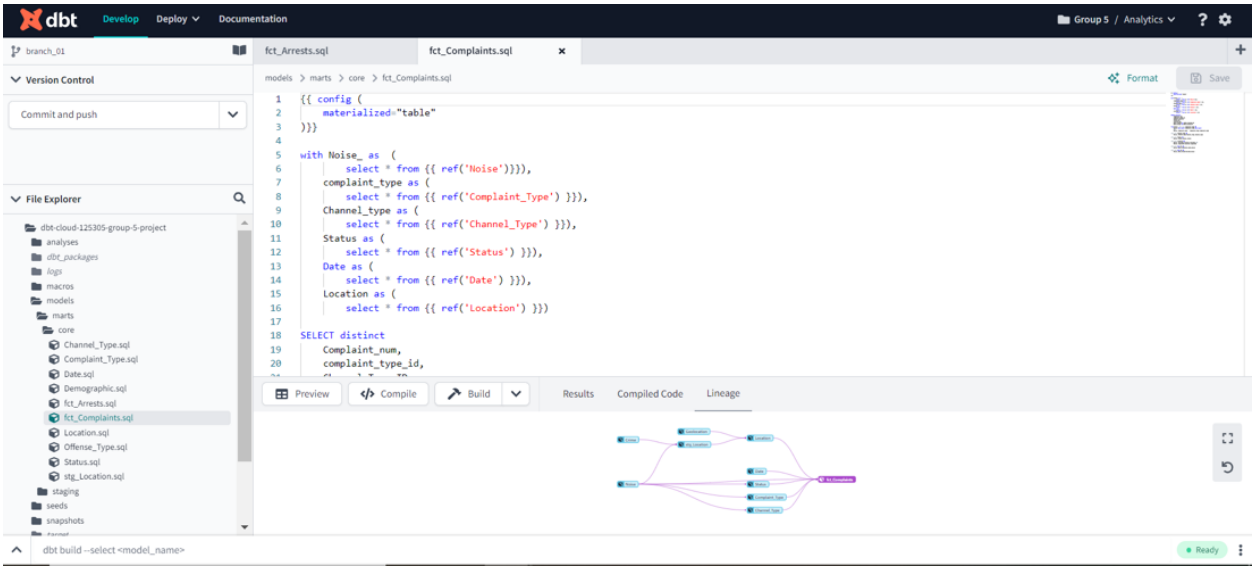
- DataProfiling-CleaningReport-Noise-Crime (file submitted with this PDF)

Summary of the report and actions performed:

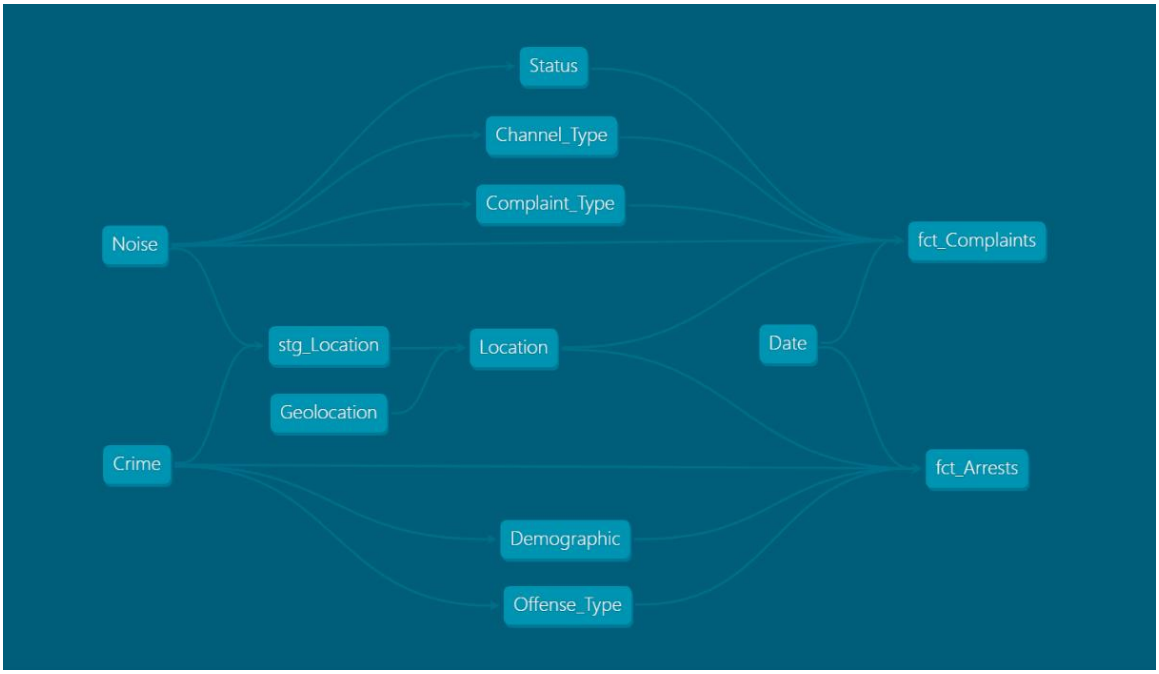
- Summary:
- Data types: all string
- Null values: Location, Incident Address, Date_closed, descriptor, Location_type, KY_CD, OFNS_DESC. Nulls value accounts for a very small portion, 1% of the data sets.
- Uniqueness: Arrest key has duplicates
- Actions:
- We changed the format of created and closed date to timestamp.

- For the Noise dataset we handled the null records for descriptor by filling with a generic description “Noise”, location_type by filling with the most occurred value. Date closed also contained null values but that is expected.
- Arrest Key in the crime data contained some duplicate values (stemming from duplicate rows) and thus we handled it in our data cleansing process by dropping duplicates.
- For the Crime dataset, we filled nulls in KY_CD with the most occurred value, and filled nulls in OFNS_DESC with NA.

Dbt Process Overview:



Lineage Graph



Lineage Graph Description:

All dimension and fact tables are seen above. Noise and Crime tables are the sources. On the top, we have the status, channel_type, and complaint_type dimensions corresponding to the Noise table. On the bottom, we have the demographic and offense_type dimensions corresponding to the Crime table. In the center we have the shared dimensions which are location and date. All dimensions merge to obtain our fact tables, fct_Complaints and fct_Arrests. As a result, we have our completed lineage map.

List of Tables

Noise Complaint Tables

1. NOISE
2. Status Dimension
3. Channel Type Dimension
4. Complaint Type Dimension

Crime Tables

5. CRIME
6. Demographics Dimension
7. Offense Type Dimension

Shared Tables

8. GEOLOCATION
9. Stage Location
10. Location Dimension
11. Date Dimension

Fact Tables

12. Complaints Fact Table
13. Arrests Fact Table

Packages

14. Package

Dbt code Description

Below we created dimensions, one staging table for location, and two fact tables. We also leveraged dbt utils to successfully create our Date dimension. We created surrogate keys with the row_number() function. For the Location dimension, we used a combination of Latitude and Longitude as a surrogate key named Location_ID. Lastly, we utilized JOINS to join the necessary dimensions to create the fact tables.

Dbt Code below:

1) NOISE: **Noise.sql**

```
SELECT
    complaint_num,
    cast(date_created as date) AS date_created,
    cast(date_closed as date) AS date_closed,
    status,
    channel_type,
    complaint_type,
    location_type,
    descriptor,
    incident_zip,
    incident_address,
    Borough,
    Latitude,
    Longitude
FROM `dw-finalproject.Noise.Noise19-22`
ORDER BY complaint_num
```

2) Status Dimension: **Status.sql**

```
with Status_ as (
select
    distinct status,
from {{ ref("Noise")}})

select
    row_number() over(order by status) as Status_ID,
    Status_.status as status
from Status_
order by Status_ID
```

3) Channel Type Dimension: **Channel_Type.sql**

```
with ChannelType as (
select
    distinct channel_type,
from {{ ref("Noise")}})

select
    row_number() over(order by channel_type) as Channel_Type_ID,
    ChannelType.channel_type as channel_type
from ChannelType
order by Channel_Type_ID
```

4) Complaint Type Dimension: **Complaint_Type.sql**

```
with ComplaintType as (
```

```
select
    distinct complaint_type,
    descriptor,
from {{ ref("Noise")}})

select
    row_number() over(order by complaint_type) as complaint_type_id,
    ComplaintType.descriptor as descriptor,
    ComplaintType.complaint_type as complaint_type
from ComplaintType
order by descriptor
```

5) CRIME: **Crime.sql**

```
SELECT
    Arrest_Num,
    cast(ARREST_DATE as date) as ARREST_DATE,
    KY_CD,
    OFNS_DESC,
    Borough,
    PRECINCT_ID,
    Age_Group,
    Sex,
    Race,
    Latitude,
    Longitude
FROM `dw-finalproject.Crime.Crime19-22`
ORDER BY ARREST_DATE DESC
```

6) Demographics Dimension: **Demographic.sql**

```
with Demographic as (
select
    distinct AGE_GROUP,
    RACE,
    SEX
from {{ ref("Crime")}})

select
    row_number() over(order by AGE_GROUP) as Demog_id,
    Demographic.AGE_GROUP as AGE_GROUP,
    Demographic.RACE as RACE,
    Demographic.SEX as SEX
from Demographic
```

7) Offense Type Dimension: **Offense_Type.sql**

```
select
    distinct KY_CD,
```

```
    OFNS_DESC
from {{ ref("Crime")}}
order by KY_CD
```

8) GEOLOCATION: **Geolocation.sql**

```
SELECT
    LatLong AS Location_ID,
    Address AS Zip_Code
FROM `dw-finalproject.Geolocation.Zip`
ORDER BY Location_ID
```

9) Stage Location: **Stg_Location.sql**

```
with crimelocation as (SELECT distinct
    Latitude AS latitude,
    Longitude AS longitude,
    concat(latitude,',',longitude) as Location_ID,
    Borough,
    PRECINCT_ID
FROM {{ ref("Crime")}}),

noiselocation as (SELECT distinct
    latitude,
    longitude,
    borough,
    incident_zip AS Zip_Code,
    incident_address,
    location_type,
    concat(latitude,',',longitude) as Location_ID
FROM {{ ref("Noise")}})

select * from
crimelocation full join noiselocation USING(Latitude, Longitude,
Location_ID, Borough)

order by Zip_Code desc
```

10) Location Dimension: **Location.sql**

```
SELECT *
FROM {{ref('stg_Location')}} FULL JOIN {{ref('Geolocation')}}
USING(Location_ID, Zip_Code)
```

11) Date Dimension: **Date.sql**

```
{{ config (
    materialized="table"
```

```
)}}

with date_spine AS (
    {{ dbt_utils.date_spine(
        datepart="day",
        start_date="cast('2019-01-01' as date)",
        end_date="cast('2022-12-31' as date)")
    }}
)
    SELECT
        cast(date_day as date) AS Date_Value,
        EXTRACT(DAY FROM date_day) AS day,                --(1 - 31)
        EXTRACT(MONTH FROM date_day) AS month,            --(1 - 12)
        EXTRACT(YEAR FROM date_day) AS year,
        EXTRACT(DAYOFWEEK FROM date_day) AS day_of_week,  --(1 - 7)

        CASE WHEN EXTRACT(DAYOFWEEK FROM date_day) > 5 THEN 'Yes'
        ELSE 'No'
        END AS Weekend
    FROM date_spine
```

12) Complaint Fact Table: **Fct_Complaints.sql**

```
{{ config (
    materialized="table"
)}}

with Noise_ as  (
        select * from {{ ref('Noise')}}),
    complaint_type as (
        select * from {{ ref('Complaint_Type') }}),
    Channel_type as (
        select * from {{ ref('Channel_Type') }}),
    Status as (
        select * from {{ ref('Status') }}),
    Date as (
        select * from {{ ref('Date') }}),
    Location as (
        select * from {{ ref('Location') }})

SELECT distinct
    Complaint_num,
    complaint_type_id,
    Channel_Type_ID,
    Status_ID,
    Location_ID,
    Date_Created AS DATE_Created_ID,
    Date_Closed AS DATE_Closed_ID

from Noise_ Left Join Complaint_Type ON
    Noise_.descriptor=Complaint_Type.descriptor
```

```
    AND
    Noise_.complaint_type =  Complaint_Type.complaint_type

Left Join Channel_Type ON
    Noise_.channel_type=Channel_Type.channel_type

Left Join Status ON
    Noise_.status=Status.status

Left Join Location ON
    Noise_.Latitude=Location.Latitude AND
    Noise_.Longitude=Location.Longitude

Left Join Date d1 ON
    Noise_.Date_Created=d1.Date_Value
Left Join Date d2 ON
    Noise_.Date_Closed=D2.Date_Value
```

13) Arrests Fact Table: **Fct_Arrests.sql**

```
{{ config (
    materialized="table"
)}}

with Crime_ as  (
        select * from {{ ref('Crime')}}),
    Demographic_ as (
        select * from {{ ref('Demographic') }}),
    Offense_type as (
        select * from {{ ref('Offense_Type') }}),
    Date as (
        select * from {{ ref('Date') }}),
    Location as (
        select * from {{ ref('Location') }})

SELECT
    Arrest_Num,
    KY_CD AS Offense_Type_ID,
    Demographic_.Demog_ID,
    Location_ID,
    ARREST_DATE AS DATE_ID

from Crime_ Left Join Demographic_ ON
    Crime_.Age_Group=Demographic_.Age_Group AND
    Crime_.Sex=Demographic_.Sex AND
    Crime_.Race=Demographic_.Race

Left Join Location ON
    Crime_.Latitude=Location.Latitude AND
    Crime_.Longitude=Location.Longitude
```

```
Left Join Date ON
    Crime_.ARREST_DATE=Date.Date_Value
```

14) Packages: **Packages.yml**

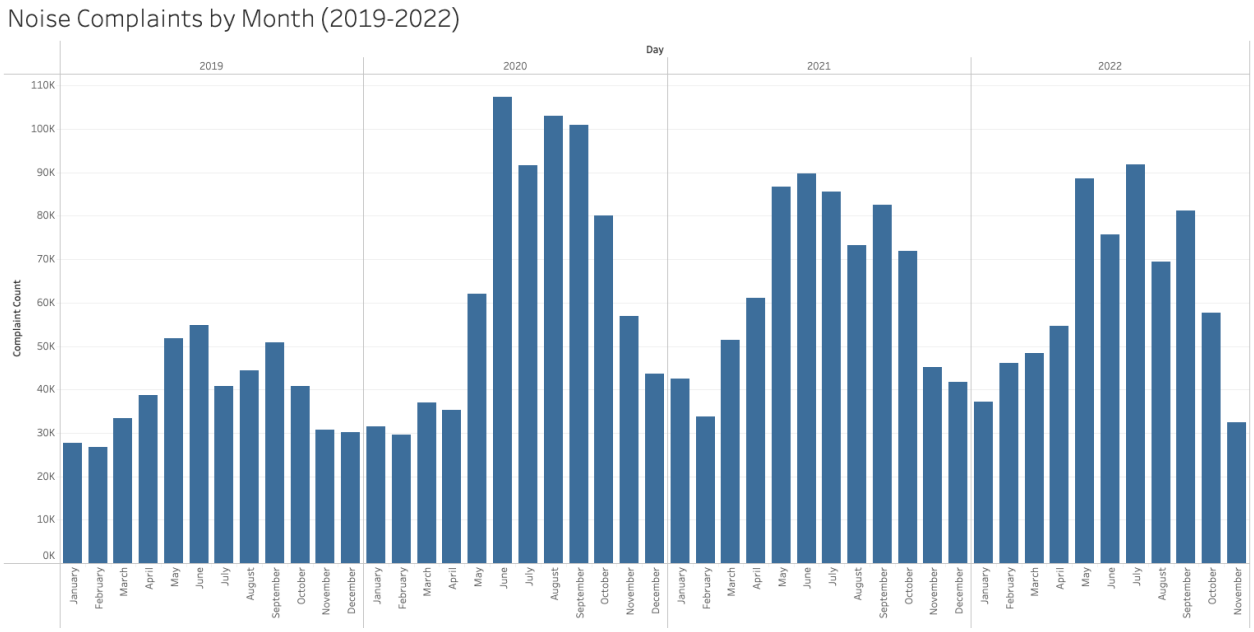
```
packages:
  - package: dbt-labs/dbt_utils
    version: 0.9.2
```

Initial Wireframe

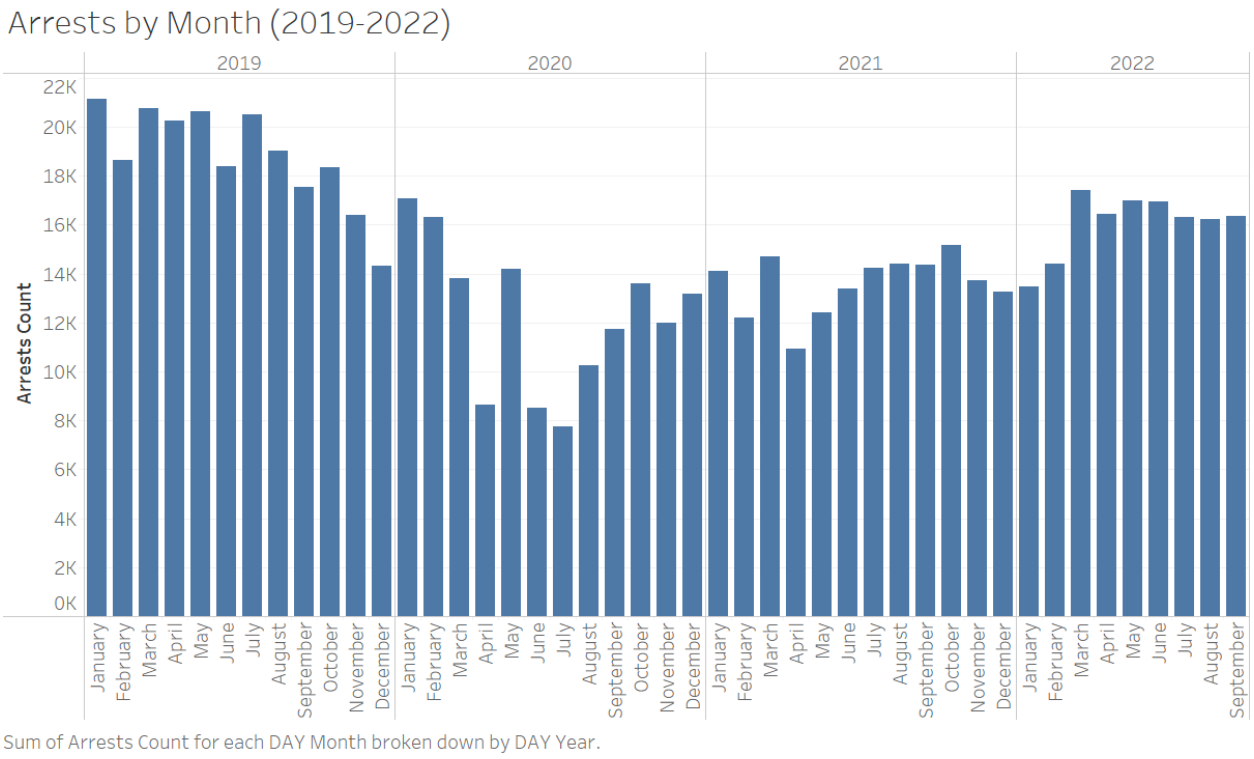
This was our initial wireframe that served as our visualization concept for how our KPIs would appear. The wireframe assisted our group by selecting appropriate visualization styles for our various KPIs.

KPI Final Visualizations

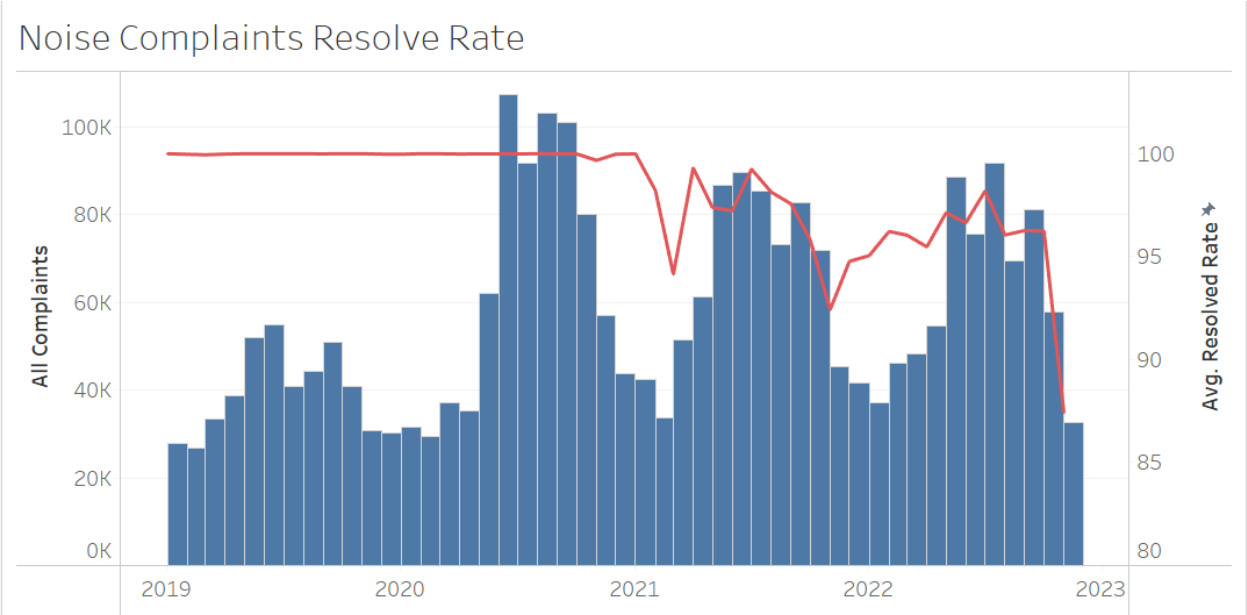
1. Noise complaints (number of complaints) by month



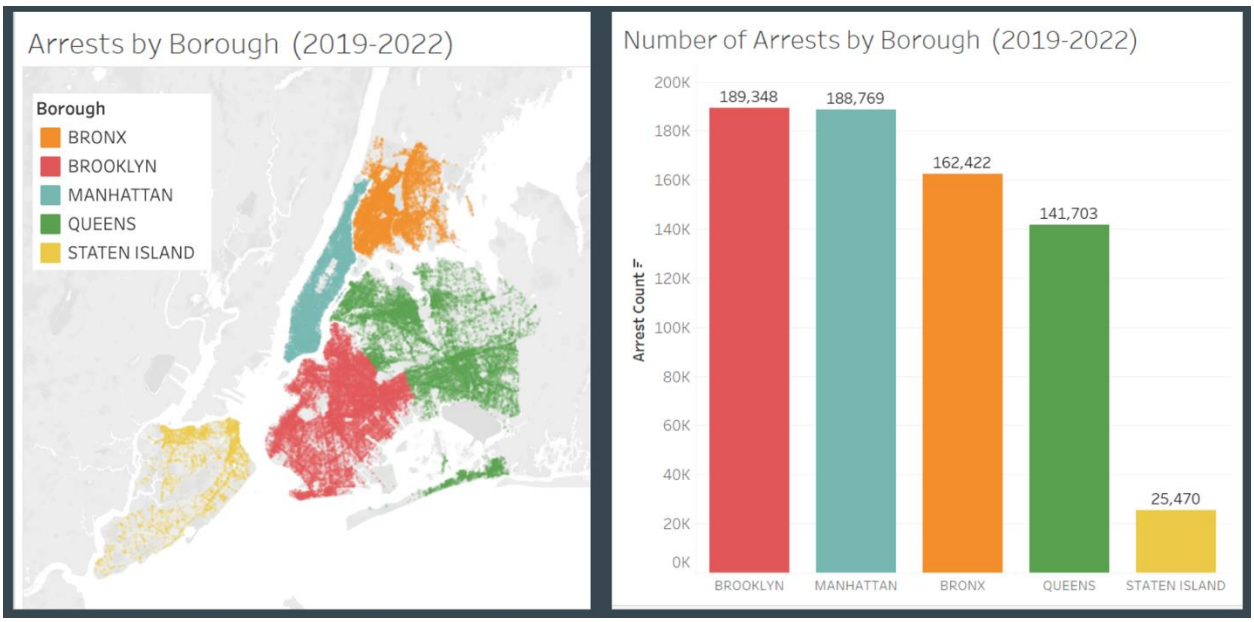
2. Crime rate (number of arrests) by month



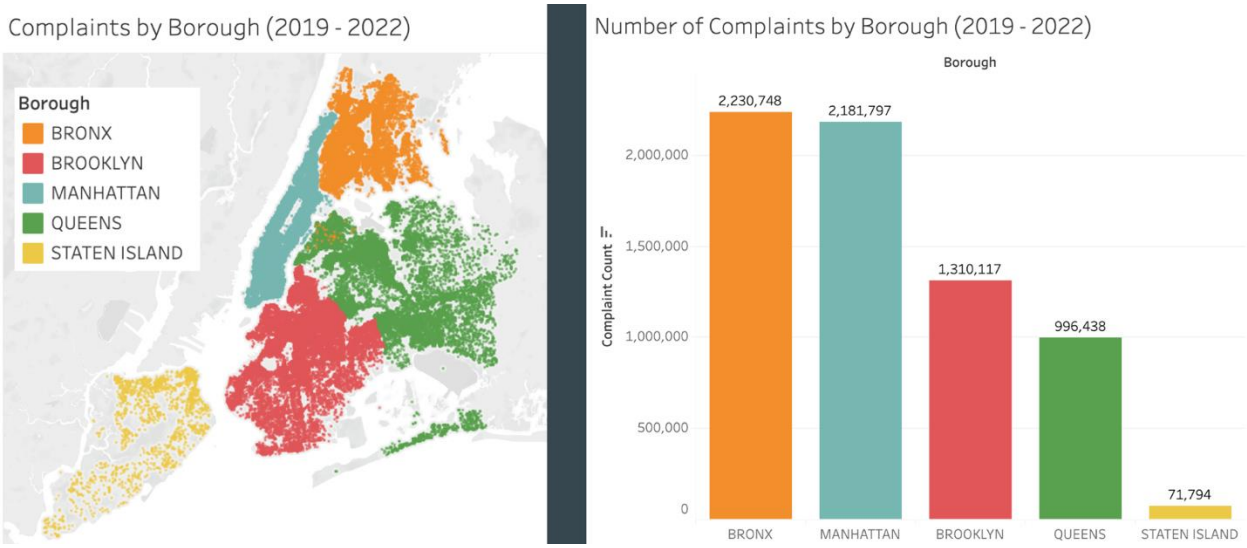
3. Noise complaint resolved rate (resolved complaints/total complaints)

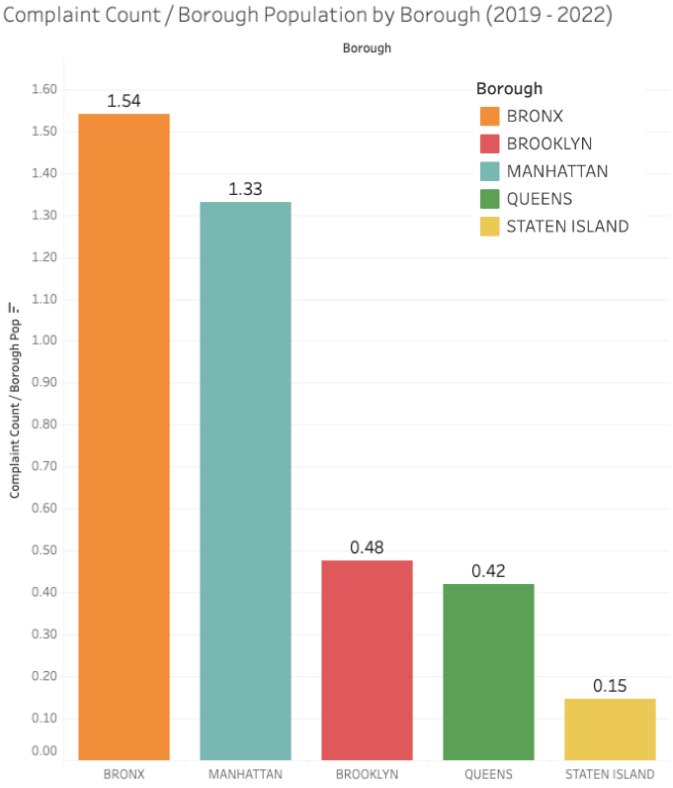


4. Number of arrests by borough

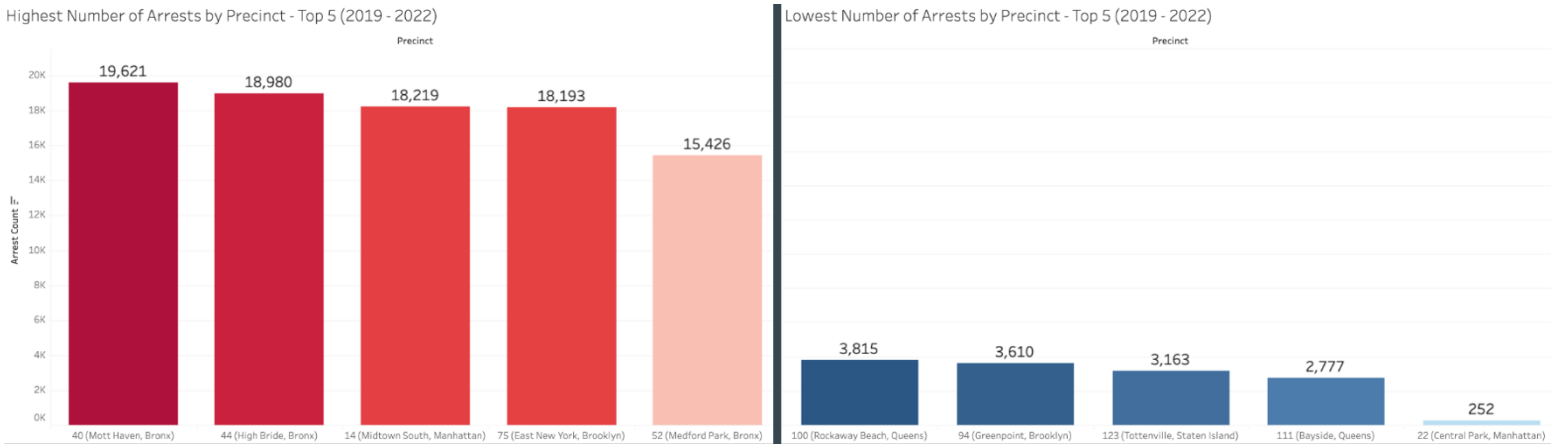


5. Noise complaints by borough and Noise Complaints / Borough Population

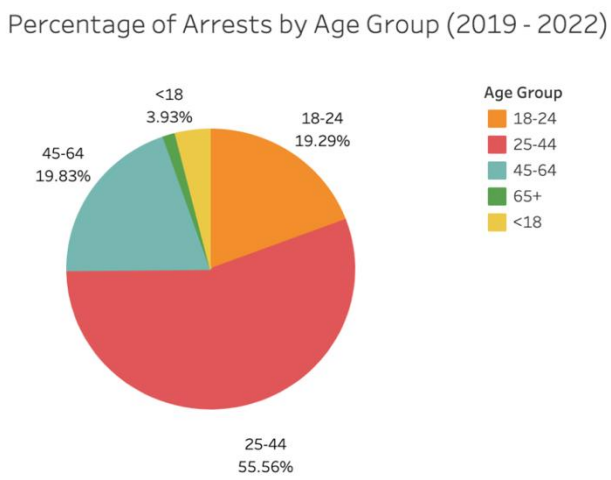


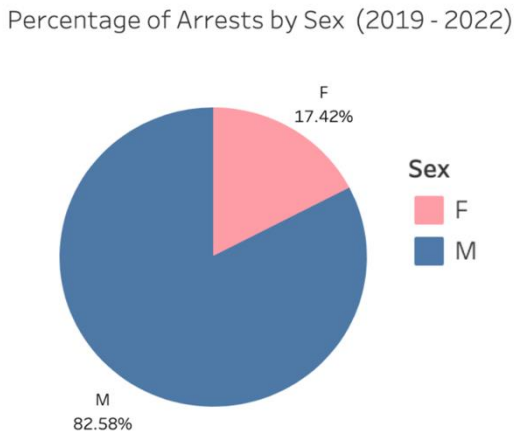


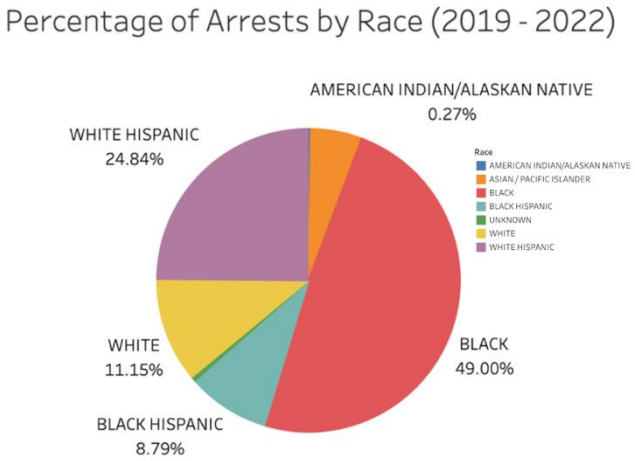
6. Highest and Lowest Number of Arrests by Precinct



7. Percentage of arrests by age group, sex, and race







Narrative Conclusions

a) the software and database tools the group used to coordinate and manage the project as well as carry out the programming tasks (list of bullet points with software or service and one sentence of what it was used for)

Our group utilized various tools to coordinate and manage this project:

- Google Big Query was used as our target DBMS tool
- Zoom was used to host all online meetings
- Shareable Microsoft Word document was used to collaborate on each deliverable and to document our progress
- Dbt was used as our ETL programming tool
- Python was utilized for data profiling
- Tableau was used as our KPI visualization tool

b) the group's experience with the project (which steps were the most difficult? Which were the easiest? what did you learn that you did not imagine you would have? if you had to do it all over again, what would you have done differently?)

The most difficult part was working collaboratively with dbt, which had limited sharing capabilities. Creating some of the dimensions and joining different tables in dbt was also challenging. The final challenge was streaming real-time data of 5M+ records from Socrata API to GCP was challenging code-wise.

Since we all had experience with visualizing data in Tableau, visualizing our KPIs was one of the easier parts of the project. Another simple part of the project was properly organizing the lineage graph, since we all had an idea of how the map should look.

Working with dbt was a skill we all did not anticipate learning in this class. We are thankful for learning more about this tool and its powerful capabilities.

If we had to do it all over again, we would have spent more time data profiling our selected datasets to ensure critical datapoints existed in both datasets.

c) if the proposed benefits can be realized by the new system.

Yes, the proposed benefits can be realized by the new system because we found that we can gain more meaningful insights by combining datasets.

Through this new system, we were able to deliver the results of the KPIs that we set out to showcase from the beginning of the project.

d) any final comments and conclusions

Our main conclusions:

- There is a relation between high noise complaints and high crime rates
- 2020 = Peak Noise Complaint Year yet Lowest Number of Arrests Year
- Queens and Staten Island both had the lowest crime rate and noise complaint rate, even though Queens is the second most populated borough (2.3M) in NYC
- Safe neighborhoods with low noise complaint rates:
    - Rockaway Beach, Queens
    - Tottenville, Staten Island
    - Bayside, Queens
- Going through all previous steps (Requirements, EDW Bus Matrix, Dimensional Model, etc.) for the ETL process was very helpful

References:

- 311 NYC Noise Complaint Data: https://data.cityofnewyork.us/Social-Services/311-Service-Requests-from-2010-to-Present/erm2-nwe9
- NYC Crime historic Data: https://data.cityofnewyork.us/Public-Safety/NYPD-Arrests-Data-Historic-/8h9b-rp9u
- NYC Crime Year to Date Data: https://data.cityofnewyork.us/Public-Safety/NYPD-Arrest-Data-Year-to-Date-/uip8-fykc
- Resource used to obtain borough population data: https://data.cityofnewyork.us/City-Government/New-York-City-Population-by-Borough-1950-2040/xywu-7bv9

Meeting Logs

Meeting 1: September 20th, 2022 // Attendance: Kevin, Thy, Raul, Arunima

- Brainstormed on which datasets to select
- Submitted initial project proposal idea

Meeting 2: October 2nd, 2022 // Attendance: Kevin, Thy, Raul, Arunima

- Reviewed Professor feedback on initial project proposal idea
- Decided to go back to the initial idea of analyzing noise complaint data and NYC crime data

Meeting 3: October 14th, 2022 // Attendance: Kevin, Thy, Raul, Arunima

- Reviewed the complaint and arrest dimensional models
- Assigned keys and organized tables
- Created bus matrix

Meeting 4: October 23rd, 2022 // Attendance: Kevin, Thy, Raul, Arunima

- Reviewed Professor feedback on Milestone #3 submission
- Added Precinct_ID to Arrest Model
- Added Location_type to 311 Noise Complaint Model
- Organized Arrest demographic data (Age, Sex, Race) into one Demographic_Dimension table
- Added KY_PD (crime classification code) and OFNS_DESC (offense description) to Arrest Model
- Merged the 311 Noise Complaint Model and NYC Arrest Model into one Integrated Datawarehouse Model

Meeting 5: November 13th, 2022 // Attendance: Kevin, Thy, Raul, Arunima

- Reviewed prior stages of project
- Selected the ETL Tool and Target DBMS (dbt and Google BigQuery)

Meeting 6: November 30th, 2022 // Attendance: Kevin, Thy, Raul, Arunima

- Successfully connected Big Query datasets with dbt
- Completed data profiling
- Removed null values

Meeting 7: December 6th, 2022 // Attendance: Kevin, Thy, Raul, Arunima

- Created tables for all dimensions in model
- Created template Fact Tables to be filled in next day

Meeting 8: December 7th, 2022 // Attendance: Kevin, Thy, Raul, Arunima

- Created Complaint and Arrest Fact Tables
- Organized Lineage Map
- Completed ETL programming

Meeting 9: December 10th, 2022 // Attendance: Kevin, Thy, Raul, Arunima

- Created visualizations for every KPI in Tableau
- Added visualizations to presentation

Meeting 10: December 12th, 2022 // Attendance: Kevin, Thy, Raul, Arunima

- Reviewed presentation
- Assigned slides to each group member to present

Meeting 11: December 16th, 2022 // Attendance: Kevin, Thy, Raul, Arunima

- Organized and formatted final report
- Finalized final report deliverables