DSBA-6520 - Project II Deliverable

Nicholas Occhipinti

https://github.com/nick-occ/dsba-6520-project

Dataset & Business Use Case

A project is funded by the NYPD to analyze data from NYC Open Data, which contains both NYPD historic complaint data and NYPD complaint data for 2021. The data includes felony, misdemeanor and violation crimes that have been reported by the NYPD. Each record in this dataset is a complaint that has occurred between 2006 to 2021. The dataset is large, containing over 7 million records. There are 35 features in the dataset containing information about the type of complaint/offense, the level of offense (felony, misdemeanor, etc), the police jurisdiction, the location of the complaint/offense such as latitude/longitude and transit district and station names if applicable.

For the purposes of the analysis the data will be filtered by dates ranging between 2019 and 2021, and by the more serious types of offenses, that being felonies. The project scope is only concerned with the NYPD, so other outside entities such as Transit Police will also be excluded. This will reduce the dataset from 7.4 million complaints to just under 270,000 records. The complaints occurred in 77 precincts within the 5 boroughs of New York City. The complaints in the datasets represent 65 categories that are more general classifications of offenses that can further be broken down into 369 more specific classifications.

The business use case is a Crime Analyst working for New York City that has been tasked to work on a project identifying patterns of recurring crime around New York City. Since New York City has such a large population of people and a large area for police to patrol, it is important to spot trends of high crime locations to better allocate police resources. Historical data can help tell a story about how areas of New York City have changed over time and the analyst can identify if these areas are seeing an increase or decrease of crime. The main goal of this project is to develop results that will aid the NYPD leadership to make better decisions and develop actionable items to help reduce crime. Some examples of improvements that this project could lead to is knowing where to increase the police presence in different areas, better communication between NYPD and communities about emerging crime patterns and identifying where to allocate funds towards technology such as adding more security cameras.

An additional use case of this project is studying how crime has changed during the COVID-19 pandemic. This was also one of the factors for the chosen date range of the data being from 2019 to 2021. This represents three different periods: pre-pandemic (2019), pandemic (2020) and vaccine/return to normal period (2021). The objective is that by analyzing these time periods, conclusions can be derived about if certain types of offenses are higher or lower than

normal which could be a result of different factors during the pandemic such as restaurants\public places being closed, reduction of tourists and people being quarantined in their home.

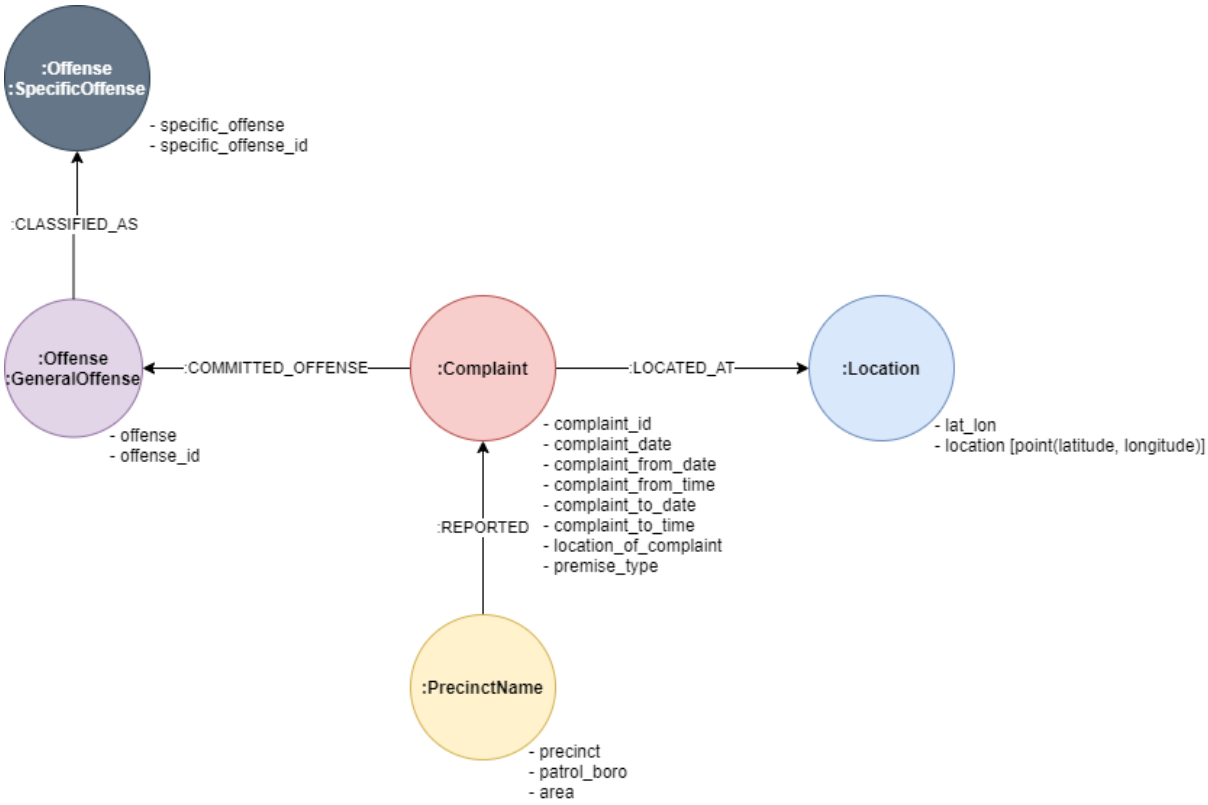
Graph Data Model Revisions

One revision made to the data model was removing the **ComplaintDate** node where the data being loaded is already aggregated by Month-Year. In place of that node type a **Complaint** node was created to hold all the complaint related properties. The reasoning for this was that the aggregation could be done through Cypher queries after the data is loaded, and by not aggregating the data initially, additional properties of the complaint can be captured. The second change was removing the **LocationName** node. Further exploratory analysis of the data showed that most of the data was not populated for these properties. There are two categories of offenses, one that is more general and the other that is more granular. The nodes are grouped together with an **Offense** label since they are related but also created under additional labels separating them into a **SpecificOffense** and a **GeneralOffense**. For the **PrecinctName** node additional spatial data was imported from https://data.cityofnewyork.us/Public-Safety/Police-Precincts/78dh-3ptz in order to capture the area each precinct covers. This allows the complaint counts used in the Cypher queries to be normalized. As a result, an **area** property was added to the **PrecinctName** node. Lastly, the properties were renamed to make them more meaningful since in the original source of the data, some of the field names were not clear without consulting the data dictionary.

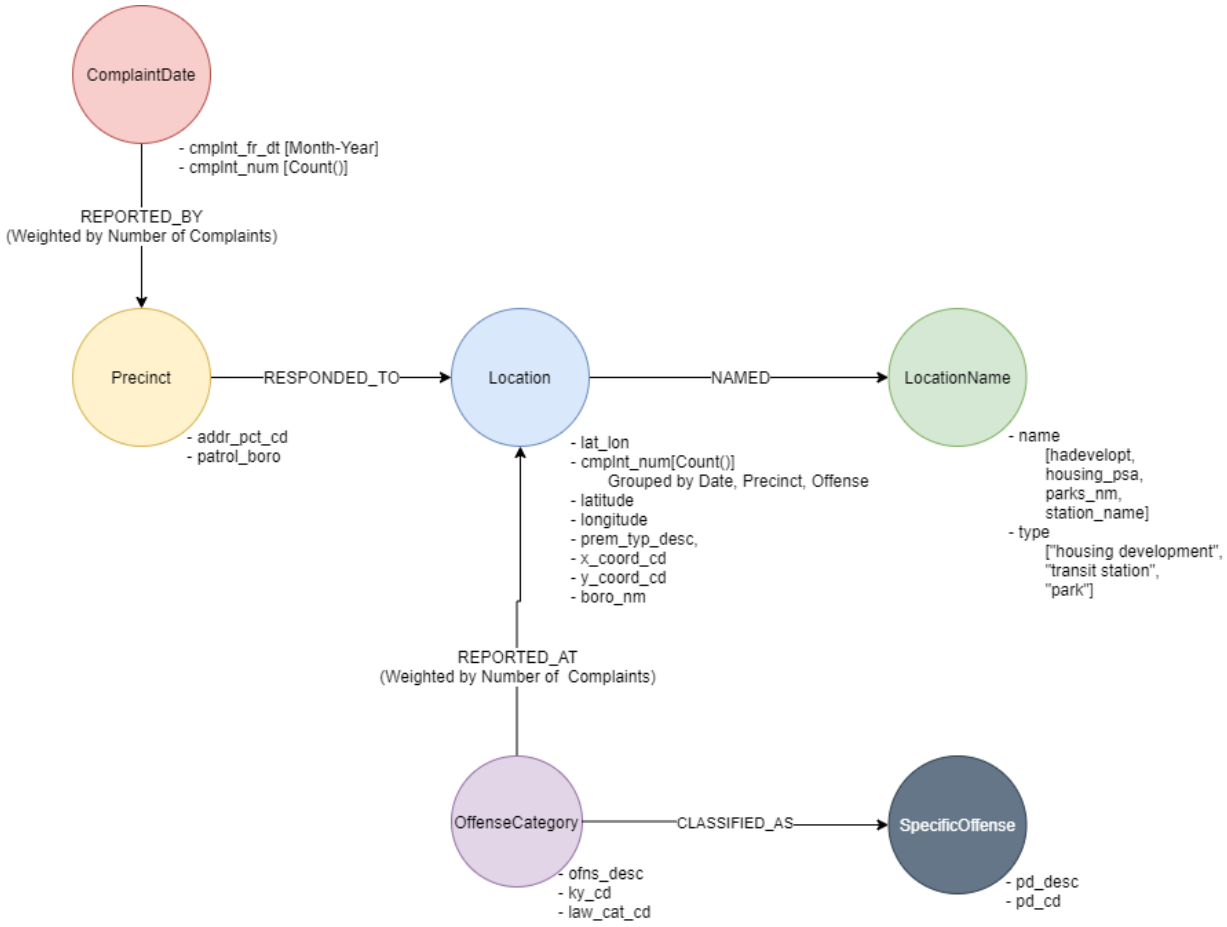
The data model now contains 5 node types, **Complaint**, **PrecinctName**, **Location**, **GeneralOffense** and **SpecificOffense**. The **Complaint** node uses the “complaint_date” property which is when the complaint was first reported and uses the "complaint_id" property for the constraint to define uniqueness. The dates are a key to determining how complaints have changed over time. The **PrecinctName** node is the precinct where the complaint was reported and is represented by the “precinct” property. The relationship between **Complaint** and **PrecinctName** is a directional relationship where the precinct reports a complaint. The **Location** node is where the complaints occur and are spatially created as points through the latitude and longitude properties which represent the smallest level of granularity in this dataset. There is a directional relationship where complaints are located at locations.

There are two types of offenses those that are more general and another category of more specific offenses which are broken up into two nodes, **GeneralOffense** and **SpecificOffense** respectively. Based on the requirements of the analysis the user could just work with the **GeneralOffense** node to get a general name of the offense, but if more information is needed, the **SpecificOffense** node offers that extra level of detail. Each offense node has an id number which is the offense code, and a description of what that offense is. There is a directional relationship between complaints where the related complaints contain details of an offense that was committed, and the general offense can get the classification of the specific offense category.

Updated Graph Data Model



Original Graph Data Model



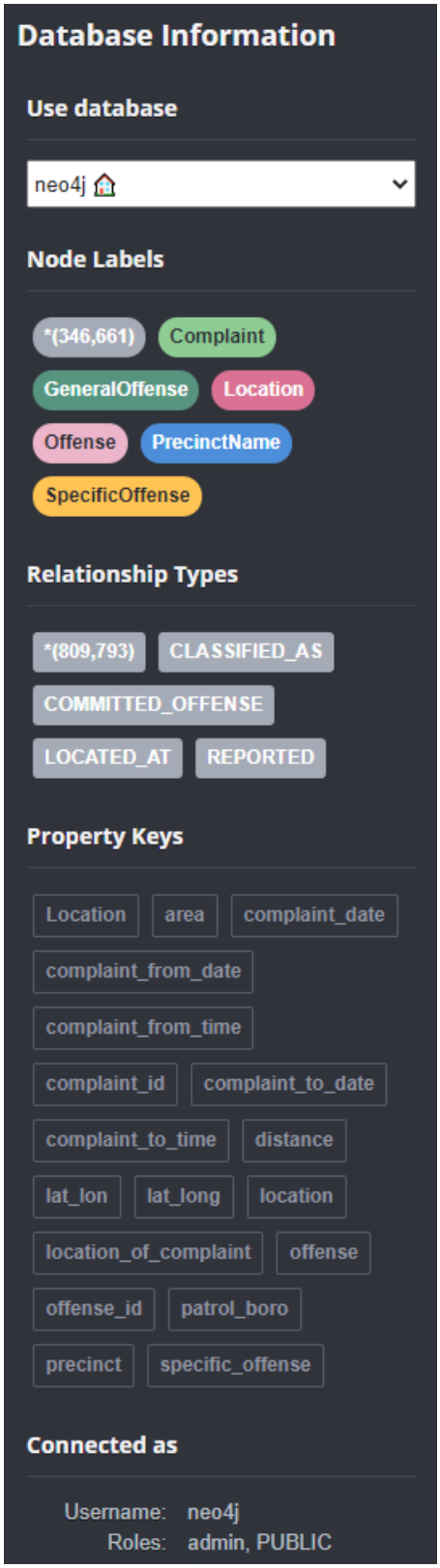
Neo4J Database Setup

Data Loading Scripts

```
// Constraints
CREATE CONSTRAINT unique_complaint ON (n:Complaint) ASSERT n.complaint_id IS UNIQUE

CREATE CONSTRAINT unique_precinct ON (n:PrecinctName) ASSERT n.precinct IS UNIQUE

CREATE CONSTRAINT unique_general_offense ON (n:GeneralOffense) ASSERT n.offense IS UNIQUE

CREATE CONSTRAINT unique_specific_offense ON (n:SpecificOffense) ASSERT n.specific_offense IS UNIQUE

CREATE CONSTRAINT unique_lat_lon ON (n:Location) ASSERT n.lat_lon IS UNIQUE

// Loading CSV's

// Load Complaint nodes
LOAD CSV WITH HEADERS FROM "file:///complaints.csv" as row
CREATE (c:Complaint {
```

```
    complaint_id: row.complaint_id,
    complaint_from_date: row.complaint_from_date,
    complaint_from_time: row.complaint_from_time,
    complaint_to_date: row.complaint_to_date,
    complaint_to_time: row.complaint_to_time,
    location_of_complaint: row.location_of_complaint,
    premise_type: row.premise_type,
    complaint_date: row.complaint_date
})

// Load PrecinctName nodes and create REPORTED relationship with Complaint
LOAD CSV WITH HEADERS FROM "file:///precincts.csv" as row
MERGE (p:PrecinctName {
    precinct: row.precinct,
    patrol_boro: row.patrol_boro,
    area: toFloat(row.area)})
WITH row, p
MATCH(c:Complaint {complaint_id: row.complaint_id})
MERGE (p)-[:REPORTED]->(c)

// Load Offense:GeneralOffense node and create COMMITTED_OFFENSE relationship with Complaint
LOAD CSV WITH HEADERS FROM "file:///offense.csv" as row
MERGE (o:Offense:GeneralOffense {
    offense: row.offense,
    offense_id: row.offense_id

})
WITH row, o
MATCH(c:Complaint {complaint_id: row.complaint_id})
MERGE (c)-[:COMMITTED_OFFENSE]->(o)

// Load Offense:SpecificOffense node and create CLASSIFIED_AS relationship with Complaint
LOAD CSV WITH HEADERS FROM "file:///specific_offense.csv" as row
MERGE (s:Offense:SpecificOffense {
    offense: row.specific_offense,
    offense_id: row.specific_offense_id

})
WITH row, s
MATCH(o:Offense {offense_id: row.offense_id})
MERGE (o)-[:CLASSIFIED_AS]->(s)

// Load Location node and create LOCATED_AT relationship with Complaint
LOAD CSV WITH HEADERS FROM "file:///location.csv" as row
// WITH row LIMIT 10
MERGE (l:Location {
location: point({
latitude:toFloat(row.latitude),
longitude:toFloat(row.longitude)}),
lat_lon: row.lat_lon
})
WITH row, l
MATCH(c:Complaint {complaint_id: row.complaint_id})
MERGE (c)-[:LOCATED_AT]->(l)
```

Database Setup



Cypher Queries

Precinct Complaints in 2021

This query captures a high level snapshot of the year to date complaints to see which precincts are dealing with the most complaints. This information will help when comparing if the hotspots of rising complaints are occurring in the precincts with the most complaints, in the future analysis that is performed at a more granular level when comparing the Location nodes. The data is also normalized to account for the area of each precinct and the number is sorted by the complaints_per_100k_sqft field. One initial finding in the data below shows that 9 out of the top 10 precincts are in Manhattan.

```
MATCH(p:PrecinctName)-[:REPORTED]->(c:Complaint)
where c.complaint_date >= "2021-01-01"
return p.precinct as precinct, p.patrol_boro as patrol_boro, count(c) as number_of_complaints,
p.area as area, count(c)/p.area * 100000 as
complaints_per_100k_sqft
order by complaints_per_100k_sqft desc
```

precinct	patrol_boro	number_of_complaints	area	complaints_per_100k_sqft
Precinct 14	MAN SOUTH	599	20510163.83	2.920503
Precinct 28	MAN NORTH	251	15289544.60	1.641645
Precinct 46	BRONX	531	38323373.38	1.385577
Precinct 33	MAN NORTH	343	25865037.87	1.326114
Precinct 32	MAN NORTH	282	23009990.36	1.225555
Precinct 6	MAN SOUTH	264	22098189.76	1.194668
Precinct 9	MAN SOUTH	255	21394233.59	1.191910
Precinct 18	MAN SOUTH	374	32261097.60	1.159291
Precinct 30	MAN NORTH	217	18845037.81	1.151497
Precinct 5	MAN SOUTH	193	18088797.95	1.066959

Top 5 Offenses per Precinct in 2021

This query gets the top 5 offenses reported by the precincts in 2021. It is not only important to understand the quantity of offenses but also what those offenses are. Part of this project is being able to give the NYPD feedback that they can use to make better decisions, and different offenses require different plans of actions. One pattern found in the example below is that Grand Larceny appears to be a common offense in most precincts.

```
MATCH(p:PrecinctName)-[:REPORTED]->(c:Complaint)-[:COMMITTED_OFFENSE]-(o:Offense)
where c.complaint_date >= "2021-01-01"
```

```
with p.precinct as precinct, p.area as area, o.offense as offense, count(c) as complaints
order by precinct, complaints desc
with precinct, area, offense, complaints, complaints/area * 100000 as complaints_per_100k_sqft
order by complaints desc
with precinct, collect({offense:offense,
complaint:complaints,complaints_per_100k_sqft:complaints_per_100k_sqft, area: area})[..5] as
top_5_offenses unwind top_5_offenses as o
return precinct, o.offense as offense, o.complaint as number_of_complaints, o.area as area,
o.complaints_per_100k_sqft as complaints_per_100k_sqft
order by precinct, complaints_per_100k_sqft desc
```

precinct	offense	number_of_complaints	area	complaints_per_100k_sqft
Precinct 1	GRAND LARCENY	143	4.731589e+07	0.302224
Precinct 1	BURGLARY	27	4.731589e+07	0.057063
Precinct 1	FELONY ASSAULT	25	4.731589e+07	0.052836
Precinct 1	ROBBERY	21	4.731589e+07	0.044383
Precinct 1	CRIMINAL MISCHIEF & RELATED OF	20	4.731589e+07	0.042269
Precinct 10	GRAND LARCENY	75	2.726732e+07	0.275055
Precinct 10	BURGLARY	35	2.726732e+07	0.128359
Precinct 10	CRIMINAL MISCHIEF & RELATED OF	21	2.726732e+07	0.077015
Precinct 10	ROBBERY	20	2.726732e+07	0.073348
Precinct 10	DANGEROUS WEAPONS	11	2.726732e+07	0.040341
Precinct 100	MISCELLANEOUS PENAL LAW	35	2.069877e+08	0.016909
Precinct 100	GRAND LARCENY	21	2.069877e+08	0.010146
Precinct 100	FELONY ASSAULT	16	2.069877e+08	0.007730
Precinct 100	THEFT-FRAUD	9	2.069877e+08	0.004348
Precinct 100	DANGEROUS WEAPONS	9	2.069877e+08	0.004348
Precinct 101	FELONY ASSAULT	55	8.467267e+07	0.064956
Precinct 101	MISCELLANEOUS PENAL LAW	49	8.467267e+07	0.057870
Precinct 101	CRIMINAL MISCHIEF & RELATED OF	34	8.467267e+07	0.040155
Precinct 101	GRAND LARCENY	26	8.467267e+07	0.030706
Precinct 101	ROBBERY	14	8.467267e+07	0.016534
Precinct 102	FELONY ASSAULT	51	1.333145e+08	0.038255
Precinct 102	MISCELLANEOUS PENAL LAW	51	1.333145e+08	0.038255
Precinct 102	GRAND LARCENY	42	1.333145e+08	0.031504
Precinct 102	GRAND LARCENY OF MOTOR VEHICLE	37	1.333145e+08	0.027754
Precinct 102	BURGLARY	29	1.333145e+08	0.021753

Difference in complaints between 2019 and 2020

It is important to understand how complaints have changed over time to determine emerging patterns of offenses. The dates for this project cover 2019, 2020, and 2021. Since 2019 and 2020 represent a full years' worth of data, this query compares the two years number of complaints and calculates the difference to see if complaints have risen, declined or stayed the same for each precinct. The other part of the use case is understanding how complaints have changed because of the pandemic, years 2019 and 2020 also represent a pre-pandemic and pandemic comparison.

```
match (p:PrecinctName)-[:REPORTED]->(c:Complaint)
with p.precinct as precinct, p.patrol_boro as patrol_boro, date(c.complaint_date).year as year,
count(c) as complaints
order by year
where year in [2019,2020]
with precinct, patrol_boro, collect(year) as years, collect(complaints) as complaints_per_year
return  precinct, patrol_boro, years, complaints_per_year, complaints_per_year[1] -
complaints_per_year[0] as difference_19_20
order by difference_19_20 desc
```

precinct	patrol_boro	years	complaints_per_year	difference_19_20
Precinct 43	BRONX	[2019,2020]	[2286,2654]	368
Precinct 104	QUEENS NORTH	[2019,2020]	[1727,1969]	242
Precinct 102	QUEENS SOUTH	[2019,2020]	[1229,1469]	240
Precinct 48	BRONX	[2019,2020]	[1970,2145]	175
Precinct 94	BKLYN NORTH	[2019,2020]	[1030,1203]	173
Precinct 33	MAN NORTH	[2019,2020]	[1042,1209]	167
Precinct 46	BRONX	[2019,2020]	[2477,2635]	158
Precinct 42	BRONX	[2019,2020]	[1768,1916]	148
Precinct 103	QUEENS SOUTH	[2019,2020]	[2098,2242]	144
Precinct 83	BKLYN NORTH	[2019,2020]	[2018,2141]	123
Precinct 40	BRONX	[2019,2020]	[2688,2789]	101
Precinct 23	MAN NORTH	[2019,2020]	[825,917]	92
...	...	...	...	...
Precinct 62	BKLYN SOUTH	[2019,2020]	[1505,1307]	-198
Precinct 5	MAN SOUTH	[2019,2020]	[1145,943]	-202
Precinct 90	BKLYN NORTH	[2019,2020]	[1827,1620]	-207
Precinct 52	BRONX	[2019,2020]	[2786,2546]	-240
Precinct 17	MAN SOUTH	[2019,2020]	[1308,1060]	-248
Precinct 101	QUEENS SOUTH	[2019,2020]	[1360,1111]	-249
Precinct 13	MAN SOUTH	[2019,2020]	[2110,1840]	-270
Precinct 71	BKLYN SOUTH	[2019,2020]	[1734,1446]	-288
Precinct 19	MAN NORTH	[2019,2020]	[2680,2385]	-295
Precinct 75	BKLYN NORTH	[2019,2020]	[4682,4344]	-338
Precinct 14	MAN SOUTH	[2019,2020]	[2608,2093]	-515
Precinct 18	MAN SOUTH	[2019,2020]	[2620,1834]	-786

Top 5 Locations per Precinct

This query aims to examine the data at a more granular level; the location of where the complaints are occurring. It provides some initial insight into potential hotspots of where complaints could be formed and can be used to verify the results of more advanced graph data science algorithms that attempt to group similar nodes into communities. This query is relevant to the project because a key part is being able to identify areas that will help show precincts where there are a high number of offenses reported. One idea is to take the JSON output of the location data and integrate it with a web mapping software to show points that are sized proportionally based on the number of complaints.

```
match(p:PrecinctName)-[:REPORTED]->(c:Complaint)-[:LOCATED_AT]->(l:Location)
with p.precinct as precinct, count(c) as complaints, l.location as location
order by precinct, complaints desc
with precinct, collect({location:location,complaint:complaints})[..5] as top_5_locations unwind
top_5_locations as l
return precinct,  l.location as location, l.complaint as number_of_complaints
order by precinct, number_of_complaints desc
```

precinct	location	number_of_complaints
Precinct 1	point({srid:4326, x:-74.00709028, y:40.72025522})	64
Precinct 1	point({srid:4326, x:-74.01060963, y:40.71009385})	56
Precinct 1	point({srid:4326, x:-73.99985714, y:40.72441101})	48
Precinct 1	point({srid:4326, x:-74.01144362, y:40.71461714})	47
Precinct 1	point({srid:4326, x:-74.00165018, y:40.72355738})	41
…	…	…
Precinct 94	point({srid:4326, x:-73.95982681, y:40.71588701})	47
Precinct 94	point({srid:4326, x:-73.95311663, y:40.72696507})	44
Precinct 94	point({srid:4326, x:-73.96278968, y:40.720082})	40
Precinct 94	point({srid:4326, x:-73.95380782, y:40.72917213})	29
Precinct 94	point({srid:4326, x:-73.95287135, y:40.72688262})	25

Graph Algorithms

Page Rank of Locations

The Page Rank centrality algorithm ranks the importance of Location nodes based on how many interactions there are with the Complaint nodes and assigns a ranking to each location and the results can then be ordered in descending order to show which locations are most central to complaints. To give the output more meaning and since the Location node is spatial, the APOC plugin's reverseGeocode function was used to pass in the latitude and longitude to get an address which is easier to understand. In the results you will notice that there are some duplicate addresses such as "Macy's 151, West 34th Street...", this is because the number of decimal places in the Location data is greater than the reverse geocoder so similar location points get grouped into the same address.

```
call gds.graph.create('nypd-comp-rank', ['Location', 'Complaint'],{
   LOCATED_AT: {}
})

CALL gds.pageRank.stream('nypd-comp-rank') YIELD nodeId, score AS pageRank
MATCH (n)<-[l:LOCATED_AT]-()
WITH gds.util.asNode(nodeId) AS n, pageRank as page_rank, count(l) as interactions
where n.location is not null
with n.location as loc, page_rank, interactions
ORDER BY page_rank DESC
call apoc.spatial.reverseGeocode(loc['y'],loc['x']) yield location
return location.description as address, page_rank
order by page_rank desc limit 20
```

address	page_rank
Macy's, 151, West 34th Street, Herald Square, ...	45.6675
New York County State Supreme Court, 111, Cent...	45.4125
Macy's, 151, West 34th Street, Herald Square, ...	36.9975
505, Gateway Drive, Brooklyn, Kings County, Ne...	27.8175
Robert F. Kennedy Bridge, Rivers Edge Road, Ma...	27.6900
Anne Fontaine, Rockefeller Plaza, Midtown, Man...	23.3550
Bloomingdale's, 1000, 3rd Avenue, Upper East S...	20.8050
5th Avenue & East 58th Street, 5th Avenue, Mid...	20.0400
1000, Sutter Avenue, East New York, Brooklyn, ...	18.7650
2, East 169th Street, The Bronx, Bronx County,...	17.8725
Bloomingdale's, 1000, 3rd Avenue, Upper East S...	17.3625
Woodhaven Boulevard, Queens Boulevard, Elmhurs...	17.2350
1011, Sutter Avenue, East New York, Brooklyn, ...	16.8525
Woodhaven Boulevard, Queens Boulevard, Elmhurs...	16.7250
NYPD 60th Precinct, 2951, West 8th Street, Wes...	16.5975
310, East 149th Street, The Bronx, Bronx Count...	16.0875
231, West 20th Street, Chelsea, Manhattan, New...	15.9600
Anne Fontaine, Rockefeller Plaza, Midtown, Man...	14.6850
The Azure, 436, Albee Square, Downtown Brookly...	14.1750
New York City Police Department - 115th Precin...	14.0475

Louvain Complaint Community Detection

The Louvain community detection algorithm identifies similar complaints and groups them into communities based on the locations they are linked to. Therefore, communities with a high number of complaints form clusters, and since they are tied to a common location we can identify where these high numbers of offenses are located. In the example below for community_id 229376 there are six complaints associated with this community and we can see they are all linked to a common location. For very large communities it becomes a valuable tool to display a cluster of complaints based on community size and then plot that cluster to the location on a map to help precincts visually see where in their area offenses are happening.

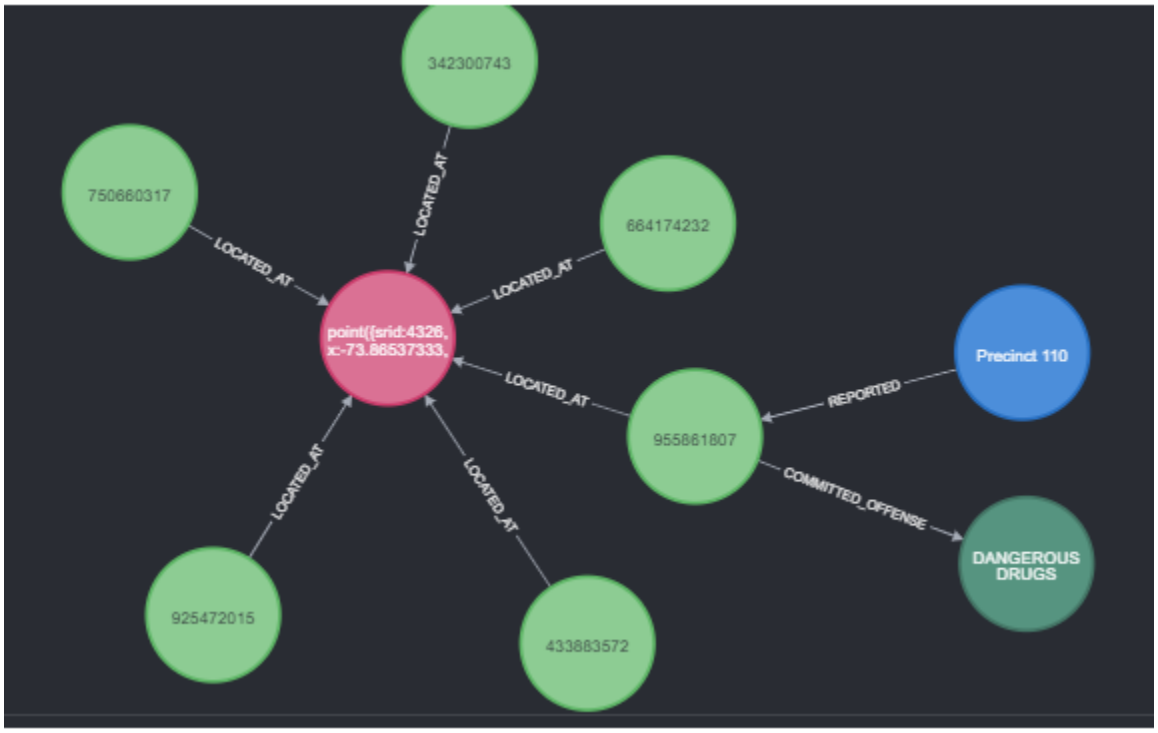
```
call gds.graph.create('nypd-comp-louvain', ['Location', 'Complaint'],{
    LOCATED_AT: {}
```

})

```
CALL gds.louvain.stream('nypd-comp-louvain') YIELD nodeId, communityId
WITH gds.util.asNode(nodeId) AS n, communityId as community_id
MATCH (n)-[l:LOCATED_AT]-(c:Complaint)
RETURN c.complaint_id as complaint, community_id
```

Example

complaint	community_id
187771337	229376
932368639	229376
428804866	229377
219333662	229377
626232189	229377
992351302	229378



Export

complaint	community_id
187771337	229376
932368639	229376
428804866	229377
219333662	229377
626232189	229377
...	...
444394680	346343
102259343	346343
550403928	346343
196345921	346344
277633413	346345

Label Propagation of Offenses

This algorithm uses Label Propagation to analyze different categories of offenses ranging from general to specific, it is useful to analyze how the offenses are related and how the different node types can be grouped together into communities. Certain groupings can be offenses that are closely related in meaning such as “Forgery,Etc..” and “Forgery”, but the added value come when the algorithm finds common relationships between two different types of offenses such as “Felony Assault” and “Robbery” and could provide valuable information on patterns of crime and how they are correlated. The previous two algorithms address the where, in terms of where patterns of high offenses are occurring while this algorithm aims to answer the what, as in what patterns of offenses are occurring.

```
CALL gds.graph.create('nypd-comp-labelprop', 'Offense', 'CLASSIFIED_AS')

CALL gds.labelPropagation.stream('nypd-comp-labelprop')
YIELD nodeId, communityId AS community_id
RETURN gds.util.asNode(nodeId).offense AS offense, community_id
ORDER BY community_id, offense
```

offense	community_id
FORGERY	269839
FORGERY,ETC.,UNCLASSIFIED-FELO	269839
ASSAULT POLICE/PEACE OFFICER	269840
FELONY ASSAULT	269840
ROBBERY	269840
...	...
MARIJUANA, POSSESSION 1, 2 & 3	269909
AGGRAVATED CRIMINAL CONTEMPT	269910
ROBBERY,LICENSED FOR HIRE VEHICLE	269911
ROBBERY,BICYCLE	269912
CONTROLLED SUBSTANCE,SALE 3	269913