

## Checkpoint 1 - Relational Analytics

MSAI 339

Prof. Jennie Rogers

October 17, 2018

Jack Richard

Quincia Hu

Ikhlas Attarwala

## Introduction

In this document we discuss answers for questions under the 'Description and Summarization' section of our project proposal. In each section, questions will include a purpose, SQL query, query result, and an analysis of the corresponding query. A recap and analysis of our description and summary can be found below.

1. What is the salary range of officers with more than a single complaint?
2. What is the distribution of allegations over the various police districts?
3. What officer ranks receive the most number of allegations?
4. How many different types of allegations are there against police officers?

## 1 Salary Distribution

### 1.1 Allegations per Officer and Corresponding Salary

The purpose of this query is to provide a view that can be used to find a correlation between salary and the number of allegations against a given officer.

#### 1.1.1 Query

```
SELECT foo.number_complaints, foo.officer, s.salary, s.year
FROM (
SELECT c.id as officer, count(t.*) as number_complaints FROM data_officerallegation t, data_officer c
WHERE t.officer_id = c.id
GROUP BY c.id
HAVING COUNT(t.*) > 1) as foo, data_salary s
WHERE s.officer_id = officer;
```

#### 1.1.2 Result

	number_complaints ↕	officer ↕	salary ↕	year ↕
1	6	1	101442	2016
2	6	1	40830	2005
3	6	1	54636	2006
4	6	1	58896	2007
5	6	1	61932	2008
6	6	1	64992	2009
7	6	1	73164	2010
8	6	1	74628	2011
9	6	1	75372	2012
10	6	1	75372	2013
11	6	1	79980	2014

### 1.1.3 Analysis

This query showed there can be multiple entries with the same salary except that year becomes an unofficial key. For each salary bonus a new entry is made. The repetition of data can be noticed on rows 9 and 10.

## 1.2 Allegations Per Officer With Distinct Salary and Correspond Years

This query handles repetitive data from the previous query. By using a distinct modifier on salary, unique years with same salaries are ignored. This reduces the amount of data in the result.

### 1.2.1 Query

```
SELECT f.number_complaints, f.officer, f.salary, s.year FROM (  
SELECT DISTINCT foo.number_complaints, foo.officer, s.salary, s.id  
FROM (  
SELECT c.id as officer, count(t.*) as number_complaints FROM data_officerallegation t, data_officer c  
WHERE t.officer_id = c.id  
GROUP BY c.id  
HAVING COUNT(t.*) > 1) as foo, data_salary s  
WHERE s.officer_id = officer) as f, data_salary s  
WHERE f.id = s.id;
```

### 1.2.2 Result

	number_complaints ↕	officer ↕	salary ↕	year ↕
1	6	1	74628	2011
2	6	1	58896	2007
3	6	1	79980	2014
4	6	1	75372	2012
5	6	1	101442	2017
6	6	1	61932	2008
7	6	1	73164	2010
8	6	1	54636	2006
9	6	1	40830	2005
10	6	1	83616	2015
11	6	1	75372	2013
12	6	1	64992	2009
13	6	1	101442	2016
14	8	2	79980	2014
15	8	2	64992	2009

### 1.2.3 Analysis

This query is a great starting point for looking into the correlation of salary to the number of complaints. By including year into this query, this also allows us to connect this result to the time at which an allegation was made against a given officer.

## 2 Police Unit Distribution

### 2.1 Allegations Per Unit ID

This query shows the number of allegations that aren't generalized to districts. The unit id's extend over into K9 units and more.

#### 2.1.1 Query

```
SELECT unit_id, count(*) as complaints
FROM
  (SELECT a.start_date as a_start,
    coalesce(a.end_date, coalesce(o.resignation_date, current_date)) as a_end,
    h.effective_date as o_start,
    coalesce(h.end_date, coalesce(o.resignation_date, current_date)) as o_end,
    a.officer_id,
    h.unit_id
  FROM data_officer allegation a, data_officerhistory h, data_officer o
  WHERE a.officer_id = h.officer_id AND o.id = a.officer_id
  ORDER BY o.id) as q1
WHERE (a_start BETWEEN o_start AND o_end) AND (a_end BETWEEN o_start AND o_end)
GROUP BY unit_id
ORDER BY complaints
```

#### 2.1.2 Result

	unit_id ↕	complaints ▼ 1
1	8	13095
2	12	11259
3	7	10299
4	5	10117
5	10	9568
6	9	9371
7	4	9248
8	6	8682
9	26	8549
10	11	7904
11	16	7628
12	3	6925
13	19	6675

#### 2.1.3 Analysis

This data showed that the units with the most allegations are in each of the districts rather than the specialty units. Once we saw there were only 25 districts in Chicago (up until 2012), we added the next query.

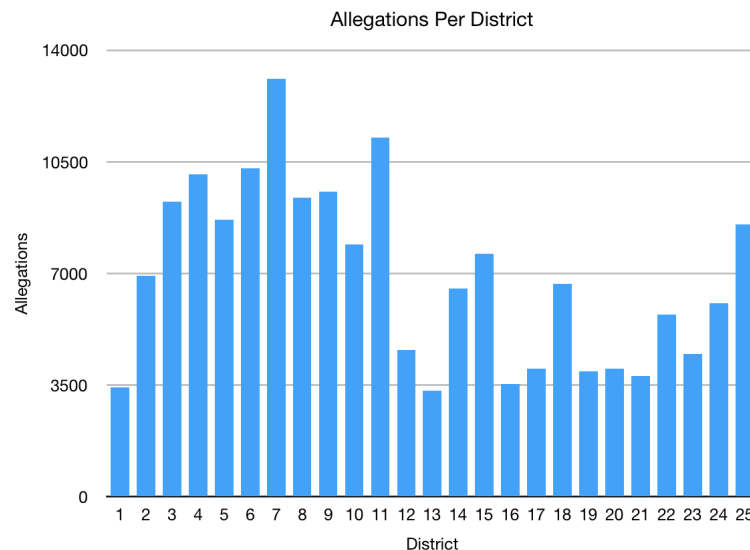
### 2.2 Allegations Per District

This query is the same as the one above, but it limits the id to the ones corresponding to the 25 stored districts.

### 2.2.1 Query

```
SELECT unit_id, count(*) as complaints
FROM
  (SELECT a.start_date as a_start,
    coalesce(a.end_date, coalesce(o.resignation_date, current_date)) as a_end,
    h.effective_date as o_start,
    coalesce(h.end_date, coalesce(o.resignation_date, current_date)) as o_end,
    a.officer_id,
    h.unit_id
  FROM data_officialallegation a, data_officerhistory h, data_officer o
  WHERE a.officer_id = h.officer_id AND o.id = a.officer_id
  ORDER BY o.id) as q1
WHERE (a_start BETWEEN o_start AND o_end) AND (a_end BETWEEN o_start AND o_end) AND unit_id BETWEEN 2 AND 26
GROUP BY unit_id
ORDER BY complaints
```

### 2.2.2 Result



### 2.2.3 Analysis

While this query is informative, it is tough to take at face value as there are no longer 25 districts, but rather 22. We will need to research how the three districts were dissolved into the others so that the data can reflect present day Chicago.

## 3 Officer Rank Allegation Distribution

### 3.1 Allegations Per Rank

Similar to the previous question, the purpose of this query is to build the foundation needed to identify any correlation between the rank of an officer and the sum total of allegations against that rank.

### 3.1.1 Query

```
SELECT c.id, count(t.*) as COMPLAINTS FROM data_officerallegation t, data_officer c  
WHERE t.officer_id = c.id  
GROUP BY c.id  
HAVING COUNT(t.*) > 1
```

### 3.1.2 Result

Field Training Officer	3893
Captain	1679
First Deputy Superintendent	67
Sergeant	36243
Deputy Superintendent	60
	477
Superintendent Of Police	21
Director Of Caps	94
Lieutenant	8567
Chief	283
Police Officer	165753
Commander	1416
Assistant Superintendent	22
Deputy Chief	536
Assistant Deputy Superintendent	38
Detective	25279
Other	26
Superintendent'S Chief Of Staff	1

### 3.1.3 Analysis

The result shows a large percentage of allegations against the police officer rank, with only a fraction of allegations being against officers above the rank of Lieutenant. Since it isn't evenly distributed, we will need to find ways to extrapolate from this query using other connections than just rank.

## 4 Allegation Categories and Subcategories

### 4.1 Allegations per Allegation Category

Understanding the allegation distribution is fundamental to the project. Knowing what types of allegations are made more often than others provides a lot of information about the types of crime we should focus on once we include the crime data in the queries.

#### 4.1.1 Query

```
SELECT c.category, count(t.*) as count  
FROM data_officerallegation t, data_allegationcategory c  
WHERE c.id = t.allegation_category_id  
GROUP BY c.category
```

### 4.1.2 Result

Category	Allegations
Unknown	1
Excessive Force	23
Racial Profiling	44
First Amendment	77
Bribery / Official Corruption	871
Drug / Alcohol Abuse	1307
Supervisory Responsibilities	4532
Domestic	6421
Criminal Misconduct	6586
Traffic	8024
Conduct Unbecoming (Off-Duty)	8138
False Arrest	8491
Verbal Abuse	13716
Lockup Procedures	14004
Illegal Search	37125
Use Of Force	56957
Operation/Personnel Violations	76546

### 4.1.3 Analysis

The most common allegations are *Operation/Personnel Violations* and *Use Of Force*. This is helpful to understand what motivates a complainant to fill out an allegation.

## 4.2 Allegations Per Allegation Subcategory

Taking the previous query one step further, understanding each subcategory provides context that may be needed when connecting crimes to resulting allegations.

### 4.2.1 Query

```
SELECT c.allegation_name, count(t.*) as count
FROM data_officer allegation t, data_allegationcategory c
WHERE c.id = t.allegation_category_id
GROUP BY c.allegation_name
```

### 4.2.2 Result

	allegation_name	count
1	Reports	7053
2	Use / Abuse Drugs / Controlled Substance – Off Duty	97
3	Intoxicated On Duty	190
4	Excessive Force / Off Duty – No Injury	6825
5	Closed Hand Strike (Punch)	66
6	Insubordination	903
7	Open Hand Strike (Slap)	23
8	Failure To Ensure Civil Rights	15
9	Arrestee-During Arrest	38
10	Traffic Pursuit	612
11	Vehicle	2
12	Arrestee-Lockup/Deletion	8
13	Assault / Battery, Etc.	115
14	Altercation / Disturbance – Traffic	524
15	Dragged	13
16	Solicit / Accept Bribe (Traffic)	141

### **4.2.3 Analysis**

The subcategory information is helpful to an extent, but there are a handful of subcategories that were either recorded incorrectly due to spelling errors or deviate significantly when compared to common subcategories.

## **5 Obstacles**

There were many obstacles to this checkpoint surrounding the intricacies of connecting tables that are multiple foreign keys apart. There are a lot of holes in the data as well. For example, many entries for the officer history are null due to either resigning or changing units.

## **6 Takeaways**

Throughout this checkpoint, we were able to gain necessary, working knowledge of the CPDP database as well as the connections between the tables. Specifically, we were able to build a great foundation for the more complex queries to come.