

My Shareable Lab Link:

<https://hub.labs.coursera.org:443/connect/sharedmyeuotyt?forceRefresh=false&path=%2F%3Ffolder%3D%2Fhome%2Fcoder%2Fproject>

Stage 1: Find and critique a dataset

1. [Choose a source of open data](#)

The data I have chosen to analyse is a log of all crimes reported to the Police Department in San Francisco, USA, from 11 May 2015 to 13 May 2015. The data is from DataSF.

2. [Assess the dataset](#)

a. Quality

This dataset was published by the San Francisco Police Department (SFPD) and the CSV file containing this information was obtained from DataSF, an official government website that compiles official open data from the City and County of San Francisco. Since its launch in 2009, the DataSF OpenData has been used by developers, analysts, residents and more to support a range of positive outcomes such as efficient city services, better decisions, and new businesses. Thus, the data is legitimate and accurate.

b. Level of detail

The dataset chosen is highly detailed and contains all the crucial attributes that are necessary to answer the questions such as the offence type, address and district where the crime occurred.

c. Documentation

This dataset contains clean data with various attributes relating to each police report such as the date, time, day of the week, category of the crime, description of the crime, the resolutions and the address of the occurrence of the crime.

d. Interrelation

The dataset can be joined with other datasets containing additional attributes such as the median tax of each district. In general, districts with lower taxes would have higher larceny and theft crimes than a district with higher taxes as the former suggests that more people are financially challenged resulting in a higher chance of resorting to theft to make ends meet. Hence, when datasets with other attributes are joined to the chosen dataset, a more insightful analysis of the crimes in San Francisco can be produced. However, the dataset must have a foreign key that can allow the tables to be joined.

e. Use

The database is used to monitor crime activity and analyse crime patterns. It may be used for predictive analysis using machine learning to predict where crimes may occur in the future. Hence, this dataset contains crucial data to ensure the safety of the people.

f. Discoverability

The dataset was easy to find as the City and County of San Francisco runs a website that compiles crucial data of San Francisco. All datasets available on this website is free and open to public. This includes the crime report logs from the SFPD, which was used in this project.

g. Terms of use

This information was created by the SFPD and the CSV file containing this information was obtained from DataSF, an official website that publishes data of the City and County of San Francisco. DataSF makes data publicly available according to open data standards and licenses datasets under the Open Data Commons Public Domain Dedication and License. Hence, the data can be trusted and used freely for personal use.

3. [Explain your interest in your report](#)

Every year, the United States of America tops the list for the country with the highest crime rates. I wanted to understand more about the crimes committed in the US, specifically San Francisco, with this dataset. I have chosen to look at data from San Francisco's crime log since it is one of the largest cities in the U.S.

Two questions I will be answering in this data exploration are:

1. *"What are the most common crimes reported in San Francisco?"*
2. *"Which district in San Francisco has the highest crime rates?"*

Knowing the most common type of crimes that takes place in the city can help the SFPD and the US government to understand what the most pressing safety concern is and implement targeted policies to tackle the issue. Thus, this can effectively lower the overall crime rate in the city. Knowing which district has the highest crime rates can help the SFPD to better protect the citizens by allocating more police officers to patrol the district and installing more surveillance cameras.

Stage 2: Model your data

1. Draw a complete E/R model of the data

Figure 1 shows the E/R model of the data.

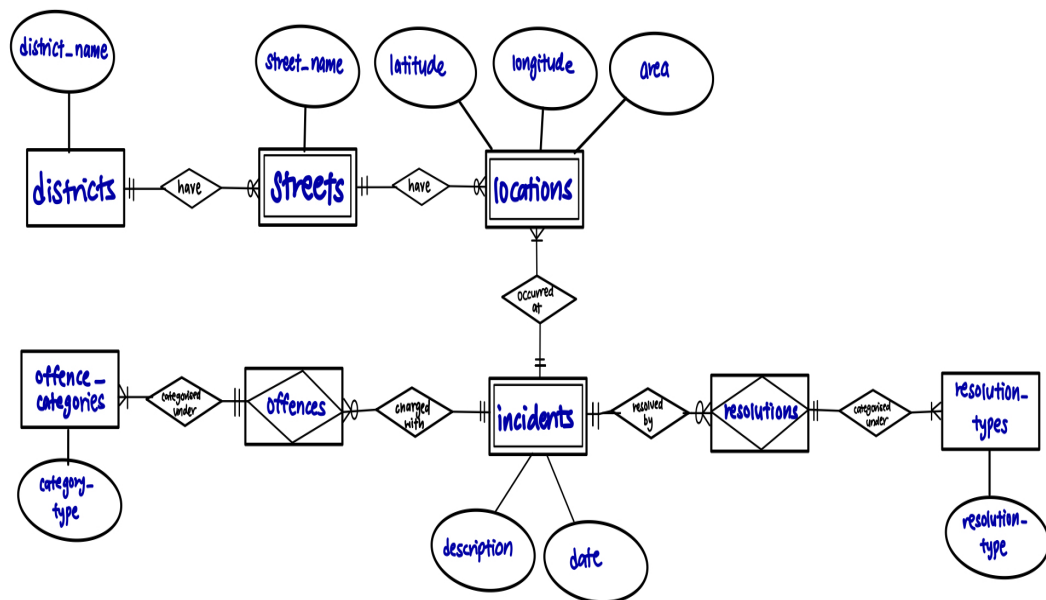


Figure 1

2. Add cardinality to the E/R diagram

Figure 2 shows how the tables are connected to each other with the use of foreign keys.

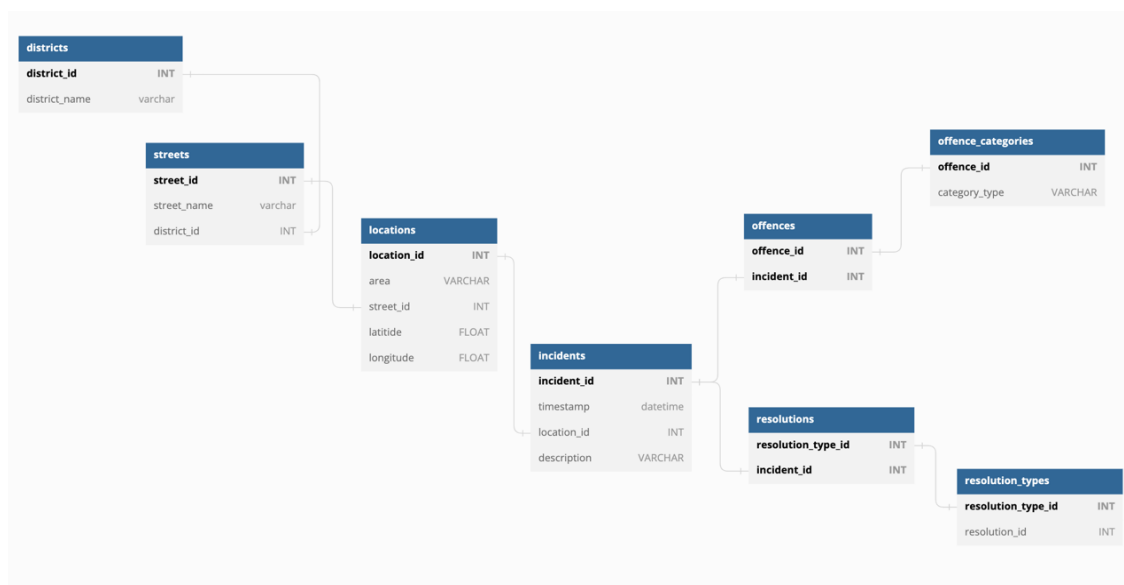


Figure 2

3. [List database tables and fields](#)

Table 1 below shows all the attributes present in the original dataset, stored in the *crimes_data* table.

Attributes	Descriptions
Timestamp	Year-Month-Day Time of the crime
Category	Type of crime
Description	Short description of the incident
the_day	The day of the week (Monday, Tuesday or Wednesday)
District	The district where the crime occurred
Resolution	How the incident was handled by the SFPD
Address	The address where the crime took place
Latitude	The X coordinate of the location where the crime took place
Longitude	The Y-coordinate of the location where the crime took place
Row_num	The ID used to represent and identify each crime log (this is unique)

Table 1

Normalisation

The tables are in normalized in fourth normalised form (4NF). However, when performing analytics task, we will be joining the tables together through the foreign keys to create a single table to perform the data analysis.

```
CREATE TABLE unnorm_data AS (  
  WITH tbl AS (  
    SELECT row_num + 1 AS row_num,  
           timestamp,  
           SUBSTRING_INDEX(  
             SUBSTRING_INDEX(address, ' / ', 1),  
             ' Block of ',  
             1  
           ) AS area,  
           SUBSTRING_INDEX(  
             SUBSTRING_INDEX(address, ' / ', -1),  
             ' Block of ',  
             -1  
           ) AS street,  
           district,  
           latitude,  
           longitude,  
           REPLACE(category, '/', ' ') AS category,  
           REPLACE(resolution, '/', '') AS resolution,  
           REPLACE(description, '/', ' /') AS description  
    FROM crimes_data  
  )  
  SELECT b.row_num,
```

```
b.timestamp,
b.area,
b.street,
b.district,
g.latitude,
g.longitude,
b.category,
b.resolution,
b.description
FROM (
    SELECT row_num,
           timestamp,
           area,
           street,
           district,
           category,
           resolution,
           description
    FROM tbl
) b
LEFT JOIN (
    SELECT area,
           street,
           district,
           AVG(latitude) AS latitude,
           AVG(longitude) AS longitude
    FROM tbl
    GROUP BY area,
           street,
           district
) g ON b.area = g.area
AND b.street = g.street
AND b.district = g.district
);
```

Stage 3: Create the database

1. [Build the database structure in MySQL](#)

All create commands are made in the mySQL console in the terminal. A database called *sf_crimes* is first created:

```
CREATE DATABASE sf_crimes;
```

Next, a database user called “USER” is created and will be granted access to the *sf_crimes* database:

```
CREATE USER 'user'@'%' IDENTIFIED WITH mysql_native_password BY  
            'calipolice';
```

Finally, a table called *crimes_data* is created to store the raw data from the csv file before inserting into the main tables as shown below in Table 2.

```
CREATE TABLE crimes_data(  
    timestamp datetime,  
    category varchar(100),  
    description varchar(255),  
    the_day varchar(10),  
    district varchar(50),  
    resolution varchar(100),  
    address varchar(255),  
    latitude float,  
    longitude float,  
    row_num int  
);
```

Once the database, table and the user have been created, the 8 main tables can be created using the CREATE commands as shown in Table 2.

CREATE commands	Description
CREATE TABLE districts (district_id int PRIMARY KEY AUTO_INCREMENT, district_name varchar(50) UNIQUE NOT NULL);	Create a table called districts with attributes district_id and district_name

<pre>CREATE TABLE streets (street_id int PRIMARY KEY AUTO_INCREMENT, street_name varchar(50) NOT NULL, district_id int NOT NULL, CONSTRAINT uc_street_district UNIQUE(street_name, district_id), FOREIGN KEY (district_id) REFERENCES districts (district_id));</pre>	<p>Create a table called streets with attributes street_id, street_name, district_id</p>
<pre>CREATE TABLE locations (location_id int PRIMARY KEY AUTO_INCREMENT, area varchar(50) NOT NULL, street_id int NOT NULL, latitude float NOT NULL, longitude float NOT NULL, CONSTRAINT uc_location UNIQUE(area, street_id, latitude, longitude), FOREIGN KEY (street_id) REFERENCES streets (street_id));</pre>	<p>Create a table called locations with attributes location_id, area, street_id, latitude and longitude</p>
<pre>CREATE TABLE incidents (incident_id int PRIMARY KEY AUTO_INCREMENT, timestamp datetime NOT NULL, location_id int NOT NULL, description varchar(255) NOT NULL, FOREIGN KEY (location_id) REFERENCES locations (location_id));</pre>	<p>Create a table called incidents with attributes timestamp, location_id and description</p>
<pre>CREATE TABLE offence_categories (category_id int PRIMARY KEY AUTO_INCREMENT,</pre>	<p>Create a table called offence_categories with attributes category_id and category_type</p>

<pre>category_type varchar(100) UNIQUE NOT NULL);</pre>	
<pre>CREATE TABLE resolution_types (resolution_type_id int PRIMARY KEY AUTO_INCREMENT, resolution_type varchar(100) UNIQUE NOT NULL);</pre>	<p>Create a table called resolution_types with attributes resolution_type_id and resolution_type</p>
<pre>CREATE TABLE offences (incident_id int, category_id int, PRIMARY KEY (incident_id, category_id), FOREIGN KEY (incident_id) REFERENCES incidents (incident_id), FOREIGN KEY (category_id) REFERENCES offence_categories (category_id));</pre>	<p>Create a tables called offences with attributes incident_id and category_id</p>
<pre>CREATE TABLE resolutions (incident_id int, resolution_type_id int, PRIMARY KEY (incident_id, resolution_type_id), FOREIGN KEY (incident_id) REFERENCES incidents (incident_id), FOREIGN KEY (resolution_type_id) REFERENCES resolution_types (resolution_type_id));</pre>	<p>Create a table called resolutions with attributes incident_id and resolution_type_id</p>

Table 2


```

        latitude,
        longitude
    FROM unnorm_data
) dt
LEFT JOIN (
    SELECT s.street_id,
           s.street_name,
           d.district_name
    FROM streets s
         LEFT JOIN districts d ON s.district_id = d.district_id
) sd ON dt.street = sd.street_name
AND dt.district = sd.district_name;

```

```

INSERT INTO incidents (incident_id, timestamp, location_id, description)
SELECT idt.row_num,
       idt.timestamp,
       lsd.location_id,
       idt.description
FROM unnorm_data idt
LEFT JOIN (
    SELECT DISTINCT lc.location_id,
                    lc.area,
                    s.street_name,
                    d.district_name
    FROM locations lc
         LEFT JOIN streets s ON lc.street_id = s.street_id
         LEFT JOIN districts d ON s.district_id = d.district_id
) lsd ON idt.area = lsd.area
AND idt.street = lsd.street_name
AND idt.district = lsd.district_name;

```

```

INSERT INTO offences (incident_id, category_id)
SELECT dt.row_num,
       o.category_id
FROM unnorm_data dt
LEFT JOIN offence_categories o ON dt.category = o.category_type;

```

```

INSERT INTO resolutions (incident_id, resolution_type_id)
SELECT dt.row_num,
       r.resolution_type_id
FROM (
    SELECT row_num,
           resolution
    FROM unnorm_data
    WHERE LOWER(resolution) NOT LIKE '%none%'
) dt
LEFT JOIN resolution_types r ON dt.resolution = r.resolution_type;

```

3. [Reflect on how well the database reflects the data](#)

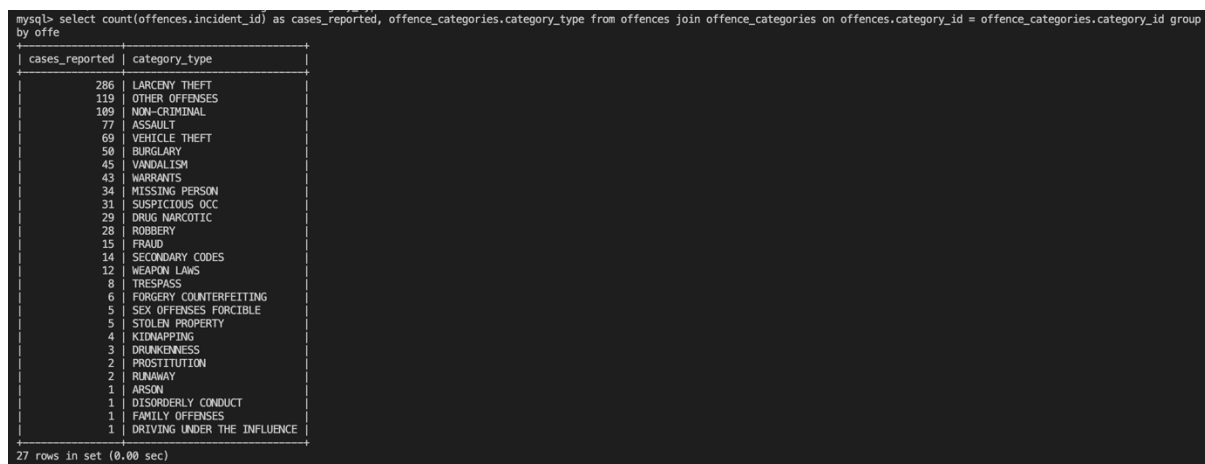
Overall, the database reflects the data well. There is no missing values or anomalies.

4. [List SQL commands that answer questions identified in the beginning of the report.](#)

Question 1: “What are the most common crimes reported?”

⇒ *SELECT COUNT(offences.incident_id) AS cases_reported,
offence_categories.category_type
FROM offences
JOIN offence_categories ON offences.category_id =
offence_categories.category_id
GROUP BY offences.category_id
ORDER BY COUNT(offences.incident_id) DESC;*

As shown in Figure 3, the most common crime is larceny theft followed by other offenses and non-criminal offenses. The least common crimes are arson, disorderly conduct, family offenses and driving under the influence.



```
mysql> select count(offences.incident_id) as cases_reported, offence_categories.category_type from offences join offence_categories on offences.category_id = offence_categories.category_id group by offence_categories.category_type order by cases_reported desc;
```

cases_reported	category_type
286	LARCENY THEFT
119	OTHER OFFENSES
109	NON-CRIMINAL
77	ASSAULT
69	VEHICLE THEFT
50	BURGLARY
45	VANDALISM
43	WARRANTS
34	MISSING PERSON
31	SUSPICIOUS OCC
29	DRUG NARCOTIC
28	ROBBERY
15	FRAUD
14	SECONDARY CODES
12	WEAPON LAWS
8	TRESPASS
6	FORGERY COUNTERFEITING
5	SEX OFFENSES FORCIBLE
5	STOLEN PROPERTY
4	KIDNAPPING
3	DRUNKENNESS
2	PROSTITUTION
2	RUNAWAY
1	ARSON
1	DISORDERLY CONDUCT
1	FAMILY OFFENSES
1	DRIVING UNDER THE INFLUENCE

27 rows in set (0.00 sec)

Figure 3

Question 2: “Which district in San Francisco has the highest crime rates?”

⇒ *SELECT COUNT(incidents.incident_id) as case_count, districts.district_name
FROM incidents
JOIN locations ON incidents.location_id = locations.location_id
JOIN streets ON streets.street_id=locations.street_id
JOIN districts ON districts.district_id = streets.district_id
GROUP BY districts.district_id*

ORDER BY case_count DESC;

The Southern district had the most crimes reported as shown in Figure 4.

```
mysql> select count(incidents.incident_id) as case_count, districts.district_name FROM incidents JOIN locations ON incidents.location_id = locations.location_id JOIN streets ON streets.street_id=locations.street_id JOIN districts ON districts.district_id = streets.district_id GROUP BY districts.district_id ORDER BY case_count DESC;
```

case_count	district_name
168	SOUTHERN
118	CENTRAL
113	INGLESTIDE
113	MISSION
104	NORTHERN
99	BAYVIEW
98	TARAVAL
68	TENDERLOIN
64	PARK
55	RICHMOND

10 rows in set (0.00 sec)

Figure 4

Stage 4: Create a simple web app

1. [Write a node.js module to present a web application that queries the database](#)

Using the web application, users can see the results of the queries created in the form of a user-friendly easy-to-read table. The contents of the table are all sorted in descending order. Figure 5 shows the screenshot of the home page of the website containing links to 2 other pages.

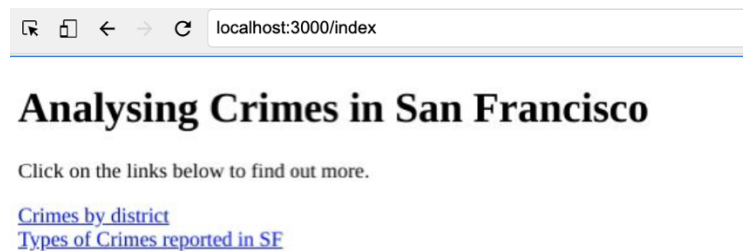


Figure 5

To run the web application, the user needs to be in the web-app directory and type “node app.js” in the terminal. This establishes a connection with the host, user, password and the *sf_crimes* database. This will run on port 3000. Querying of the database will be done in app.js and the web application will be made using express and render templates using mustache-express to dynamically change presentation views. (See Figure 6 below)

```

const express = require('express');
const bodyParser = require('body-parser');
const mysql = require('mysql');
const mustacheExpress = require('mustache-express');

const app = express();
const port = 3000;

app.engine('html', mustacheExpress());
app.set('view engine', 'html');
app.set('views', './templates');
app.use(bodyParser.urlencoded({ extended: true }));
app.listen(port, function () {
  console.log("Example app listening on port " + port);
})

//connect to the database
var dbcon = mysql.createConnection({
  host: 'localhost',
  user: 'sanfran',
  password: 'calipolice',
  database: 'sf_crimes'
})

function templateRenderer(template, res) {
  return function (error, results, fields) {
    if (error)
      throw error;
    res.render(template, { data: results });
  }
}

```

Figure 6

A web-app template is created for the home page in html as shown in Figure 7.

```

//In app.js
//HOME PAGE (index)
app.get('/index', function (req, res) {
  dbcon.query("select count(incidents.incident_id)as case_count, districts.district_name
FROM incidents
JOIN locations ON incidents.location_id = locations.location_id
JOIN streets ON streets.street_id=locations.street_id
JOIN districts ON districts.district_id = streets.district_id
GROUP BY districts.district_id
ORDER BY case_count DESC;",
  templateRenderer('index', res)
);
})

```

```

mid-term > crime-records > web-app > templates > <> index.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <style>
5      table {
6          border-collapse: collapse;
7          width: 100%;
8      }
9
10     th,
11     td {
12         padding: 8px, 16px;
13         border: 1px solid #ccc;
14     }
15
16     th {
17         background: #eee;
18     }
19 </style>
20
21
22 <head>
23     <meta charset="utf-8" />
24     <title> 2015 Crime Records in San Francisco, California</title>
25 </head>
26
27 <body>
28     <h1>Analysing Crimes in San Francisco</h1>
29     <p>Click on the links below to find out more.</p>
30     <div class="top">
31         <a href="/area">Crimes by district</a>
32         <br>
33         <a href="/types">Types of Crimes reported in SF</a>
34     </div>
35
36
37 </body>
38
39 </html>

```

Figure 7

When the user clicks on “Crimes by District”, the web application displays a page (localhost:3000/area) containing a table with the number of crimes reported per district in San Francisco in descending order. This allows users to see which district is the least safe as shown in Figure 9.

```

//In app.js
//DISTRICT (area)

```

```

app.get('/area', function (req, res) {
    dbcon.query("SELECT COUNT(incidents.incident_id) as
case_count,districts.district_name
FROM incidents
JOIN locations ON incidents.location_id = locations.location_id
JOIN streets ON streets.street_id=locations.street_id
JOIN districts ON districts.district_id = streets.district_id
GROUP BY districts.district_id
ORDER BY case_count DESC;",
    templateRenderer('area', res)
    );
})

```

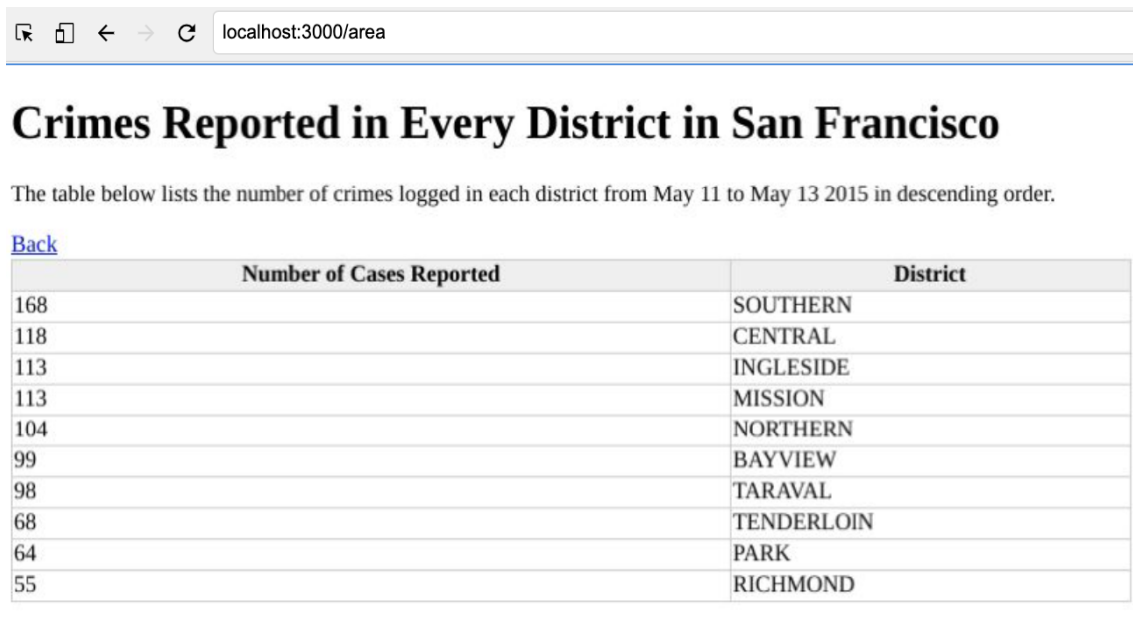
area.html is created to display the information queried as shown in Figures 8 and 9.

```

<? area.html x
mid-term > crime-records > web-app > templates > <? area.html > <? html > <? body > <? table > <? tr > <? td
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <style>
5     table {
6         border-collapse: collapse;
7         width: 100%;
8     }
9     th,
10    td {
11        padding: 8px, 16px;
12        border: 1px solid #ccc;
13    }
14    th {
15        background: #eee;
16    }
17 </style>
18
19 <head>
20     <meta charset="utf-8" />
21     <title> 2015 Crime Records in San Francisco, California</title>
22 </head>
23
24 <body>
25     <h1>Crimes Reported in Every District in San Francisco</h1>
26     <p>The table below lists the number of crimes logged in each district from May 11 to May 13 2015 in descending order.</p>
27     <div class="top">
28         <a href="/index">Back</a>
29     </div>
30
31     <table>
32         <tr>
33             <th>Number of Cases Reported</th>
34             <th>District</th>
35         </tr>
36         <tr>
37             <td>{{#data}}
38                 <td>{{case_count}}</td>
39                 <td>{{district_name}}</td>
40             </tr>
41             <td>{{/data}}
42         </table>
43     </body>
44 </html>

```

Figure 8



The browser address bar shows 'localhost:3000/area'. The page title is 'Crimes Reported in Every District in San Francisco'. Below the title, a paragraph states: 'The table below lists the number of crimes logged in each district from May 11 to May 13 2015 in descending order.' A blue link labeled 'Back' is positioned to the left of the table. The table has two columns: 'Number of Cases Reported' and 'District'. The data is as follows:

Number of Cases Reported	District
168	SOUTHERN
118	CENTRAL
113	INGLESIDE
113	MISSION
104	NORTHERN
99	BAYVIEW
98	TARAVAL
68	TENDERLOIN
64	PARK
55	RICHMOND

Figure 9

When the “Types of Crimes Report in SF” is clicked from the home page (localhost:3000/types), it displays a table containing information on all the different offence types that were reported in San Francisco, along with the number of cases reported for each offence type. The list is ranked from highest to lowest number of reports for an offence type. This allows users to easily view the different types of offences that occurred in the city and determine the most commonly occurring offences.

```
//In app.js
//OFFENCE TYPES (types)
app.get('/types', function(req, res) {
  dbcon.query("SELECT COUNT(offences.incident_id) as cases_reported,
    offence_categories.category_type
  FROM offences
  JOIN offence_categories ON offences.category_id = offence_categories.category_id
  GROUP BY offences.category_id
  ORDER BY COUNT(offences.incident_id) DESC; ",
    templateRenderer('types', res)
  );
})
```

Types.html is created to display the information as queried. See Figures 10 and 11.


```

< area.html      < types.html X
mid-term > crime-records > web-app > templates > < types.html > html > style > table
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <style>
5      table {
6          border-collapse: collapse;
7          width: 100%;
8      }
9      th, td {
10         padding: 8px, 16px;
11         border: 1px solid #ccc;
12     }
13     th {
14         background: #eee;
15     }
16 </style>
17
18 <head>
19     <meta charset="utf-8"/>
20     <title> 2015 Crime Records in San Francisco, California</title>
21 </head>
22
23 <body>
24     <h1>Ranking of Offence Types Reported</h1>
25     <p>The table below lists the offence types reported in San Francisco City during the period of May 11th 2015 to May 13th 2015.</p>
26     <p>The list is ranked from highest to lowest number of reports for each offence type.</p>
27     <div class="top">
28         <a href="/index">Back</a>
29     </div>
30
31     <table>
32         <tr>
33             <th>Cases Reported</th>
34             <th>Offence Type</th>
35         </tr>
36         {{#data}}
37         <tr>
38             <td>{{cases_reported}}</td>
39             <td>{{category_type}}</td>
40         </tr>
41         {{/data}}
42     </table>
43 </body>
44
45 </html>

```

Figure 10

Ranking of Offence Types Reported

The table below lists the offence types reported in San Francisco City during the period of May 11th 2015 to May 13th 2015.

The list is ranked from highest to lowest number of reports for each offence type.

[Back](#)

Cases Reported	Offence Type
286	LARCENY THEFT
119	OTHER OFFENSES
109	NON-CRIMINAL
77	ASSAULT
69	VEHICLE THEFT
50	BURGLARY
45	VANDALISM
43	WARRANTS
34	MISSING PERSON
31	SUSPICIOUS OCC
29	DRUG NARCOTIC
28	ROBBERY
15	FRAUD
14	SECONDARY CODES
12	WEAPON LAWS
8	TRESPASS
6	FORGERY COUNTERFEITING
5	SEX OFFENSES FORCIBLE
5	STOLEN PROPERTY
4	KIDNAPPING
3	DRUNKENNESS
2	PROSTITUTION
2	RUNAWAY
1	ARSON

Figure 11

References

<https://data.sfgov.org/Public-Safety/Police-Department-Incident-Reports-Historical-2003/tmnf-yvry>

<https://www.databasestar.com/database-normalization/>

https://teams.microsoft.com/_#/pdf/viewer/teams/https%3A~2F~2Fmymailsimedu-my.sharepoint.com~2Fpersonal~2Fsreenuch001_mymail_sim_edu_sg~2FDocument%2F%5BCM3010%5D%20Databases%20%2526%20Advanced%20Data%20Techniques~2FMid-Term%20Coursework%20Example.pdf?threadId=19:aomY70C77zYTEq2MScSzGTEx7YAhoJ2bBBIhBxt9X9w1@thread.tacv2&messageId=1665907721111&baseUrl=https%3A~2F~2Fmymailsimedu-my.sharepoint.com~2Fpersonal~2Fsreenuch001_mymail_sim_edu_sg&fileId=E9B425C5-45F1-49E0-AD92-DD5EAA572A1C&ctx=chiclet&viewerAction=view

<https://datacatalog.worldbank.org/public-licenses>

<https://dbdiagram.io/home>