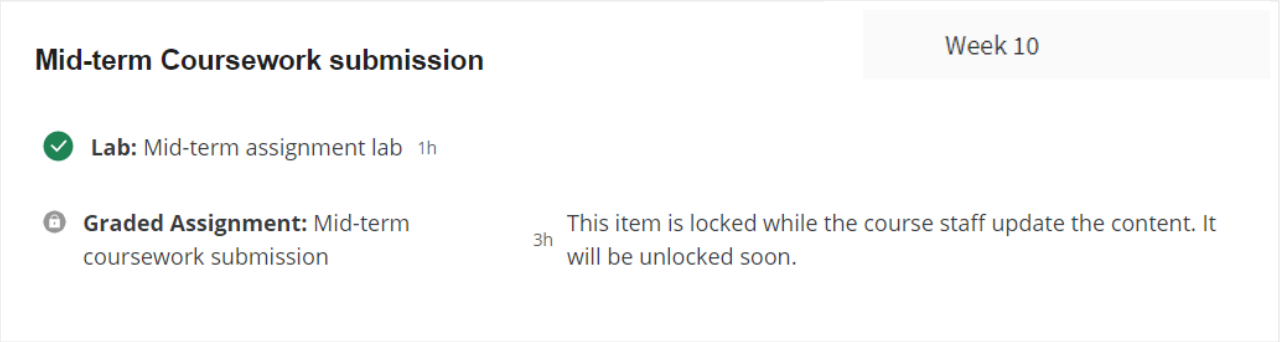
Databases and Advanced Data Techniques

Tarapong Sreenuch

Mid-Term Coursework: Crime Watch Use Case

Pre-Requisites

1: Week 10's Coursera Lab Environment



2: `SQL`, `node.js` and HTML Files. These are where all the code snippets are contained.

SQL: `load-data.sql, initiate-data-tables.sql, ingest-data.sql`
Node.js: `web-app-intro.js, express-and-mysql.js`
HTML: `index.html`

Pre-Requisites

3: Relational Schema + MySQL Export Tools

Drawing Tool: `https://dbdiagram.io`
Shared Diagram: `https://dbdiagram.io/d/619f3eae02cf5d186b678a5d`

4: Entity-Relationship Diagram Drawing Tool

Drawing Tool: `https://erdplus.com`
Shared Diagram: `clf-crime-data-erd-diagram.png`

5: MS Team Shared Files

Folder URL: `https://mymailsimedu.sharepoint.com/:f:/r/sites/DatabasesAdvDataTech2021-22SemesterOct-Mar/Shared%20Documents/General/clf-crime-watch?csf=1&web=1&e=ATRYyK`

Setting a Context: Crime Watch Use Case

- We are part of Volunteer Group in San Francisco. The initiative is to make the public more inform about safety and crime within the community.

- We are tasked to utilise public datasets and create a data-driven application that can inform the public on various crime related aspects.
  - These can include, but not limited to, crime rates, types of crime, crime hotspots and police performance in resolving the crimes.

Setting a Context: Crime Watch Use Case (cont.)

The project roadmap contains 2 deliverables:

1. PoC of data-driven application (which is the scope of our Mid-Term Assignment)
   - The PoC application is to base on 10-year historical crime data (*static CSV data*) occurred in San Francisco.
   - The focus of this development phase is on
     1. *Data Exploration* (Completeness of the Open Data)
     2. *Database Design* (ER diagram and Relational Schema)
     3. *Underlying Data Tables* (Populated with the Historical Data)
     4. *SQL queries* (Data Insights or Statistics)
     5. *Web App* (with Database Connection and Simple Information Retrieval)

Setting a Context: Crime Watch Use Case (cont.)

2. Initial Release Web App (not in the scope of this assignment)
    - The application is to base live (near real-time) data accessible through *APIs* or daily published *CSV files*, which is to be ingested into our underly database.
    - The release is to be more complete and visually appealing.

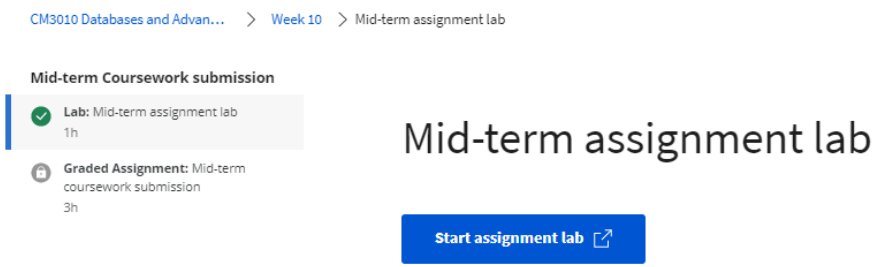
Setting a Context: Crime Watch Use Case (cont.)

- What kind of data do we need for the PoC? *(Our expectation on the data)*
  - Crime Location and Date & Time
    - Granularity: Block, Street, Hour
  - Nature of Crimes
    - Granularity: Type
  - Resolutions
    - Granularity: Type

These will be used to justify if your data is complete or not.

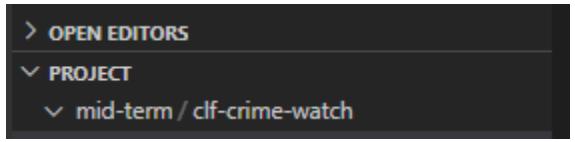
Before We Start

Step 1: cd Launch Coursera Lab Environment for our Mid-Term assignment.



Step 2: On the `bash` terminal, create new directories called '`mid-term`' and 'clf-crime-watch', where '`clf-crime-watch`' is a subdirectory of '`mid-term`'.

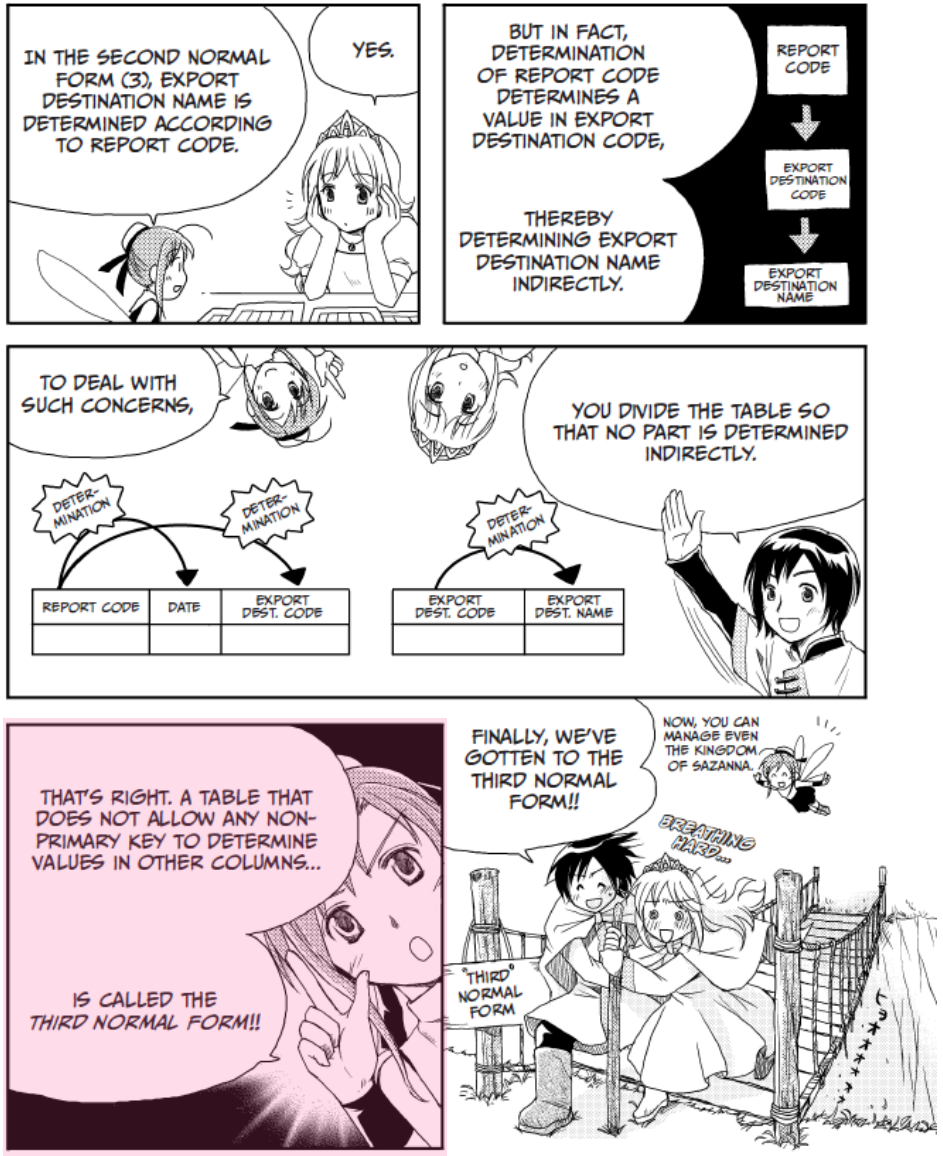
```
$ mkdir mid-term
$ mkdir mid-term/clf-crime-watch
```



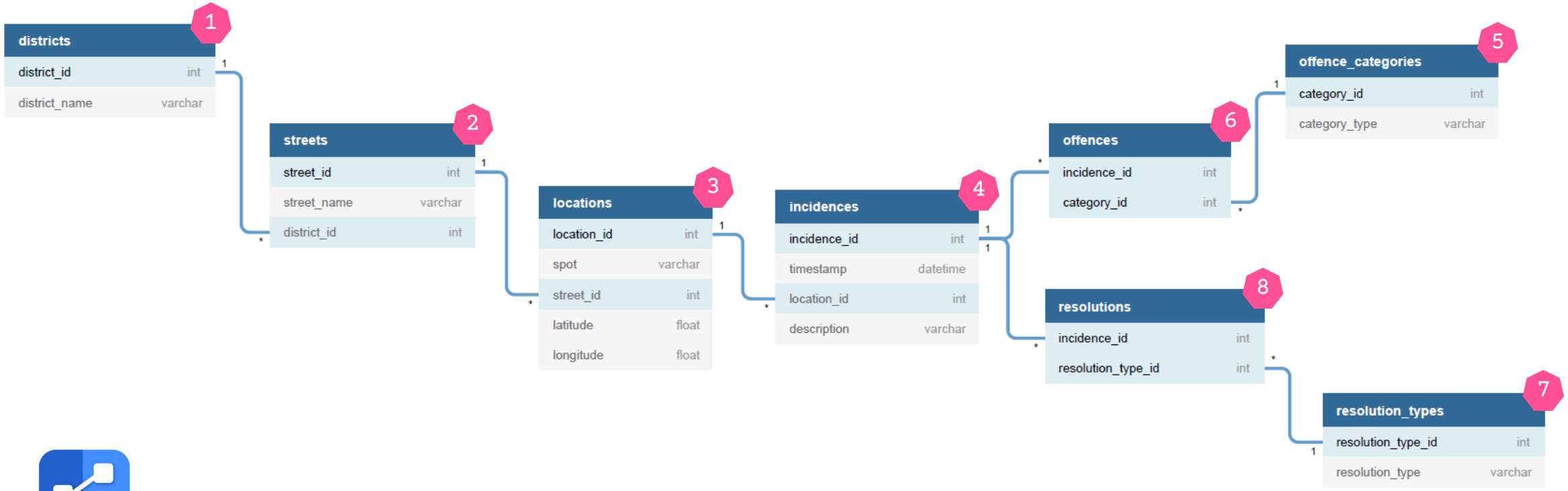
Step 3: cd into the new directory.

```
$ cd mid-term/clf-crime-watch
```

3rd Normal Form



Relational Schemas



https://dbdiagram.io

MySQL in Coursera Lab Environment

Step 1: Open a terminal and launch MySQL console

```
$ mysql
```

Step 2: Create a new database called ‘`clf_crime_data`’ and use the newly created database.

```
> CREATE DATABASE clf_crime_data;
> USE clf_crime_data;
```

Step 3: Create a database user called ‘`francis`’ and grant access to ‘`clf_crime_data`’ database.

```
> CREATE USER 'francis'@'%' IDENTIFIED WITH mysql_native_password BY 'california’;
> GRANT ALL ON clf_crime_data.* TO 'francis'@'%';
```

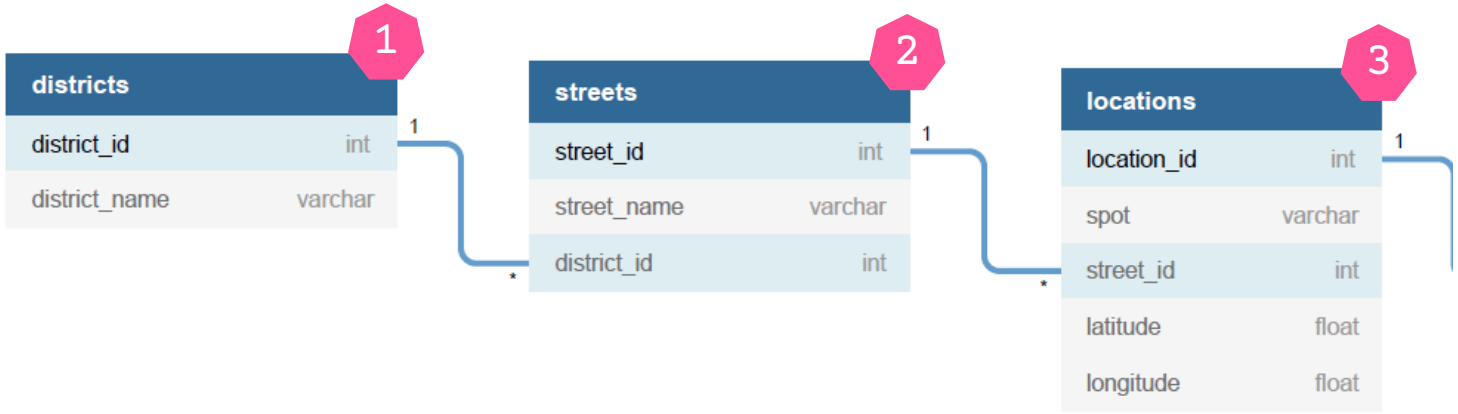
Table Creation

1

```sql
CREATE TABLE districts (
  district_id int PRIMARY KEY AUTO_INCREMENT,
  district_name varchar(32) UNIQUE NOT NULL
);
```

3

```sql
CREATE TABLE locations (
  location_id int PRIMARY KEY AUTO_INCREMENT,
  spot varchar(32) NOT NULL,
  street_id int NOT NULL,
  latitude float NOT NULL,
  longitude float NOT NULL
  CONSTRAINT uc_location UNIQUE(spot, street_id, latitude, longitude)
);
ALTER TABLE locations ADD FOREIGN KEY (street_id) REFERENCES streets (street_id);
```



2

```sql
CREATE TABLE streets (
  street_id int PRIMARY KEY AUTO_INCREMENT,
  street_name varchar(32) NOT NULL,
  district_id int NOT NULL,
  CONSTRAINT uc_street_district UNIQUE(street_name, district_id)
);
ALTER TABLE streets ADD FOREIGN KEY (district_id) REFERENCES districts (district_id);
```

Table Creation (cont.)

4

incidences	
incidence_id	int
timestamp	datetime
location_id	int
description	varchar

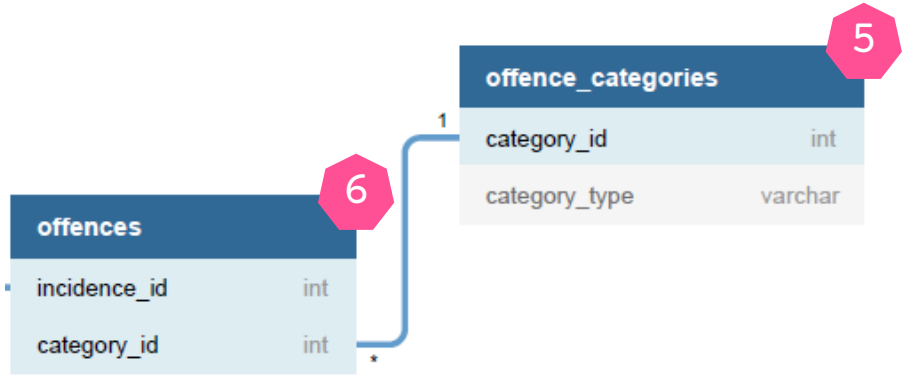
4

```
CREATE TABLE incidences (
  incidence_id int PRIMARY KEY AUTO_INCREMENT,
  timestamp datetime NOT NULL,
  location_id int NOT NULL,
  description varchar(255) NOT NULL,
  CONSTRAINT uc_incidence UNIQUE(timestamp, location_id, description)
);
ALTER TABLE incidences ADD FOREIGN KEY (location_id) REFERENCES locations (location_id);
```

Table Creation (cont.)

5

```sql
CREATE TABLE offence_categories (
  category_id int PRIMARY KEY AUTO_INCREMENT,
  category_type varchar(64) UNIQUE NOT NULL
);
```



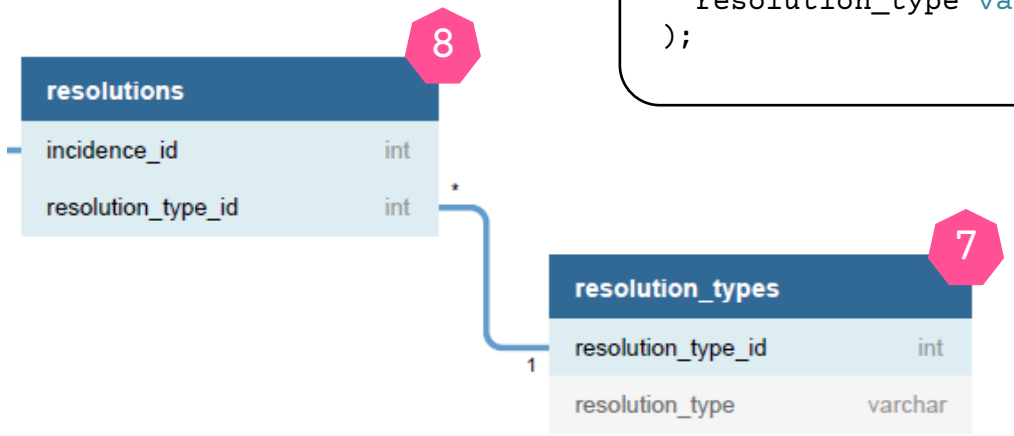
6

```sql
CREATE TABLE offences (
  incidence_id int NOT NULL,
  category_id int NOT NULL,
  PRIMARY KEY (incidence_id, category_id)
);
ALTER TABLE offences ADD FOREIGN KEY (incidence_id) REFERENCES incidences (incidence_id);
ALTER TABLE offences ADD FOREIGN KEY (category_id) REFERENCES offence_categories (category_id)
```

Table Creation (cont.)

7

```
CREATE TABLE resolution_types (
  resolution_type_id int PRIMARY KEY AUTO_INCREMENT,
  resolution_type varchar(64) UNIQUE NOT NULL
);
```



8

```
CREATE TABLE resolutions (
  incidence_id int,
  resolution_type_id int,
  PRIMARY KEY (incidence_id, resolution_type_id)
);
ALTER TABLE resolutions ADD FOREIGN KEY (incidence_id) REFERENCES incidences (incidence_id);
ALTER TABLE resolutions ADD FOREIGN KEY (resolution_type_id) REFERENCES resolution_types (resolution_type_id);
```

Data Loading: *'raw_data'* Table

In [1]:
```
import pandas as pd

# load a csv file to Pandas' dataframe for initial exploration and formating
df = pd.read_csv('sf_crime_data.csv')
df.head()
```

Out[1]:

	Dates	Category	Descript	DayOfWeek	PdDistrict	Resolution	Address	X	Y
0	2015-05-13 23:53:00	WARRANTS	WARRANT ARREST	Wednesday	NORTHERN	ARREST, BOOKED	OAK ST / LAGUNA ST	-122.425892	37.774599
1	2015-05-13 23:53:00	OTHER OFFENSES	TRAFFIC VIOLATION ARREST	Wednesday	NORTHERN	ARREST, BOOKED	OAK ST / LAGUNA ST	-122.425892	37.774599
2	2015-05-13 23:33:00	OTHER OFFENSES	TRAFFIC VIOLATION ARREST	Wednesday	NORTHERN	ARREST, BOOKED	VANNESS AV / GREENWICH ST	-122.424363	37.800414
3	2015-05-13 23:30:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Wednesday	NORTHERN	NONE	1500 Block of LOMBARD ST	-122.426995	37.800873
4	2015-05-13 23:30:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Wednesday	PARK	NONE	100 Block of BRODERICK ST	-122.438738	37.771541

In [2]:
```
# remove ',' from the column values, 'Category' and 'Resolution' in particular, and replace it with '/' character
# if a column value contains ',', then MySQL will fail to load the data.
# MySQL cannot distinguish between ',' contained in the column and the one used to separate the columns.
df['Resolution'] = df['Resolution'].apply(lambda x: x.replace(',', '/'))
df['Descript'] = df['Descript'].apply(lambda x: x.replace(',', '/'))

# add index column
df['RowNum'] = range(len(df))

# preview re-formatted data
df.head()
```

Out[2]:

	Dates	Category	Descript	DayOfWeek	PdDistrict	Resolution	Address	X	Y	RowNum
0	2015-05-13 23:53:00	WARRANTS	WARRANT ARREST	Wednesday	NORTHERN	ARREST/ BOOKED	OAK ST / LAGUNA ST	-122.425892	37.774599	0
1	2015-05-13 23:53:00	OTHER OFFENSES	TRAFFIC VIOLATION ARREST	Wednesday	NORTHERN	ARREST/ BOOKED	OAK ST / LAGUNA ST	-122.425892	37.774599	1
2	2015-05-13 23:33:00	OTHER OFFENSES	TRAFFIC VIOLATION ARREST	Wednesday	NORTHERN	ARREST/ BOOKED	VANNESS AV / GREENWICH ST	-122.424363	37.800414	2
3	2015-05-13 23:30:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Wednesday	NORTHERN	NONE	1500 Block of LOMBARD ST	-122.426995	37.800873	3
4	2015-05-13 23:30:00	LARCENY/THEFT	GRAND THEFT FROM LOCKED AUTO	Wednesday	PARK	NONE	100 Block of BRODERICK ST	-122.438738	37.771541	4

In [3]:
```
# export formatted data to a csv file, sf-crime-data.csv
df.to_csv('sql-loadable-clf-crime-data.csv', index=False)

# create a subset of data, e.g. 1st 1,000 rows, for development and testing of SQL codes
df.iloc[:1000].to_csv('tiny-clf-crime-data.csv', index=False)
```

Load Data Into Coursera Lab Environment

Step 1: Ensure that we are inside the ‘mid-term/clf-crime-watch/’ folder.

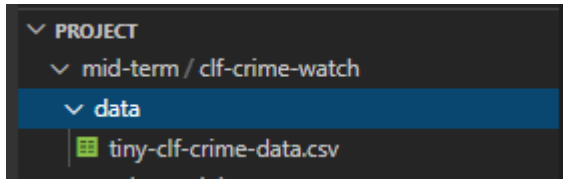
```
$ pwd
```

```
coder@db76968d0f6e:~/project/mid-term/clf-crime-watch$ pwd
/home/coder/project/mid-term/clf-crime-watch
```

Step 2: Create a new directory called ‘data’, the folder will be a subdirectory under ‘mid-term/clf-crime-watch/’.

```
$ mkdir data
```

Step 3: Using mouse to drag the csv file, e.g. `tiny-clf-crime-data.csv`, from our local directory and drop inside our Coursera Lab Environment.



Data Loading: *'raw_data'* Table

2

```
LOAD DATA INFILE '/home/coder/project/mid-term/clf-crime-watch/data/tiny-clf-crime-data.csv'
INTO TABLE clf_crime_data.raw_data
FIELDS TERMINATED BY ','
ENCLOSED BY ''
LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

1

```
CREATE TABLE raw_data (
  timestamp datetime,
  category varchar(64),
  description varchar(255),
  day_of_week varchar(16),
  district varchar(32),
  resolution varchar(64),
  address varchar(128),
  latitude float,
  longitude float,
  row_num int
);
```

```
mysql> SELECT * FROM raw_data LIMIT 10;
+---------------------+---------------+-----------------------------+------------+----------+---------------+-----------------------------+----------+-----------+---------+
| timestamp           | category      | description                 | day_of_week | district | resolution    | address                     | latitude | longitude | row_num |
+---------------------+---------------+-----------------------------+------------+----------+---------------+-----------------------------+----------+-----------+---------+
| 2015-05-13 23:53:00 | WARRANTS      | WARRANT ARREST              | Wednesday  | NORTHERN | ARREST/ BOOKED | OAK ST / LAGUNA ST          | -122.426 |   37.7746 |       0 |
| 2015-05-13 23:53:00 | OTHER OFFENSES | TRAFFIC VIOLATION ARREST    | Wednesday  | NORTHERN | ARREST/ BOOKED | OAK ST / LAGUNA ST          | -122.426 |   37.7746 |       1 |
| 2015-05-13 23:33:00 | OTHER OFFENSES | TRAFFIC VIOLATION ARREST    | Wednesday  | NORTHERN | ARREST/ BOOKED | VANNESS AV / GREENWICH ST   | -122.424 |   37.8004 |       2 |
| 2015-05-13 23:30:00 | LARCENY/THEFT | GRAND THEFT FROM LOCKED AUTO | Wednesday  | NORTHERN | NONE           | 1500 Block of LOMBARD ST    | -122.427 |   37.8009 |       3 |
| 2015-05-13 23:30:00 | LARCENY/THEFT | GRAND THEFT FROM LOCKED AUTO | Wednesday  | PARK     | NONE           | 100 Block of BRODERICK ST   | -122.439 |   37.7715 |       4 |
| 2015-05-13 23:30:00 | LARCENY/THEFT | GRAND THEFT FROM UNLOCKED AUTO | Wednesday  | INGLESIDE | NONE           | 0 Block of TEDDY AV         | -122.403 |   37.7134 |       5 |
| 2015-05-13 23:30:00 | VEHICLE THEFT | STOLEN AUTOMOBILE           | Wednesday  | INGLESIDE | NONE           | AVALON AV / PERU AV         | -122.423 |   37.7251 |       6 |
| 2015-05-13 23:30:00 | VEHICLE THEFT | STOLEN AUTOMOBILE           | Wednesday  | BAYVIEW  | NONE           | KIRKWOOD AV / DONAHUE ST    | -122.371 |   37.7276 |       7 |
| 2015-05-13 23:00:00 | LARCENY/THEFT | GRAND THEFT FROM LOCKED AUTO | Wednesday  | RICHMOND | NONE           | 600 Block of 47TH AV        | -122.508 |   37.7766 |       8 |
| 2015-05-13 23:00:00 | LARCENY/THEFT | GRAND THEFT FROM LOCKED AUTO | Wednesday  | CENTRAL  | NONE           | JEFFERSON ST / LEAVENWORTH ST | -122.419 |   37.8078 |       9 |
+---------------------+---------------+-----------------------------+------------+----------+---------------+-----------------------------+----------+-----------+---------+
10 rows in set (0.02 sec)
```

Data Cleaning: *'denorm_table'* Table

```
CREATE TABLE dnorm_data
  AS ( WITH tbl AS ( SELECT row_num + 1 AS row_num,
                            timestamp,
                            SUBSTRING_INDEX(SUBSTRING_INDEX(address, ' / ', 1), ' Block of ', 1) AS spot,
                            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 raw_data
       )

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

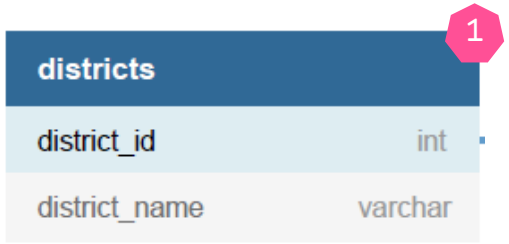
Data Cleaning: *'dnorm_data'* Table

```
mysql> SELECT * FROM dnorm_data LIMIT 25;
```

row_num	timestamp	spot	street	district	latitude	longitude	category	resolution	description
1	2015-05-13 23:53:00	OAK ST	LAGUNA ST	NORTHERN	-122.42588806152344	37.77459716796875	WARRANTS	ARREST BOOKED	WARRANT ARREST
2	2015-05-13 23:53:00	OAK ST	LAGUNA ST	NORTHERN	-122.42588806152344	37.77459716796875	OTHER OFFENSES	ARREST BOOKED	TRAFFIC VIOLATION ARREST
3	2015-05-13 23:33:00	VANNESS AV	GREENWICH ST	NORTHERN	-122.42436218261719	37.8004150390625	OTHER OFFENSES	ARREST BOOKED	TRAFFIC VIOLATION ARREST
4	2015-05-13 23:30:00	1500	LOMBARD ST	NORTHERN	-122.42699432373047	37.800872802734375	LARCENY THEFT	NONE	GRAND THEFT FROM LOCKED AUTO
5	2015-05-13 23:30:00	100	BRODERICK ST	PARK	-122.43873596191406	37.771541595458984	LARCENY THEFT	NONE	GRAND THEFT FROM LOCKED AUTO
6	2015-05-13 23:30:00	0	TEDDY AV	INGLESIDE	-122.40325164794922	37.71343231201172	LARCENY THEFT	NONE	GRAND THEFT FROM UNLOCKED AUTO
7	2015-05-13 23:30:00	AVALON AV	PERU AV	INGLESIDE	-122.42332458496094	37.72513961791992	VEHICLE THEFT	NONE	STOLEN AUTOMOBILE
8	2015-05-13 23:30:00	KIRKWOOD AV	DONAHUE ST	BAYVIEW	-122.37127685546875	37.72756576538086	VEHICLE THEFT	NONE	STOLEN AUTOMOBILE
9	2015-05-13 23:00:00	600	47TH AV	RICHMOND	-122.50819396972656	37.7765998840332	LARCENY THEFT	NONE	GRAND THEFT FROM LOCKED AUTO
10	2015-05-13 23:00:00	JEFFERSON ST	LEAVENWORTH ST	CENTRAL	-122.4190902709961	37.80780029296875	LARCENY THEFT	NONE	GRAND THEFT FROM LOCKED AUTO
11	2015-05-13 22:58:00	JEFFERSON ST	LEAVENWORTH ST	CENTRAL	-122.4190902709961	37.80780029296875	LARCENY THEFT	NONE	PETTY THEFT FROM LOCKED AUTO
12	2015-05-13 22:30:00	0	ESCOLTA WY	TARAVAL	-122.48798370361328	37.737667083740234	OTHER OFFENSES	NONE	MISCELLANEOUS INVESTIGATION
13	2015-05-13 22:30:00	TURK ST	JONES ST	TENDERLOIN	-122.41241455078125	37.78300476074219	VANDALISM	NONE	MALICIOUS MISCHIEF / VANDALISM OF VEHICLES
14	2015-05-13 22:06:00	FILLMORE ST	GEARY BL	NORTHERN	-122.43291473388672	37.78435516357422	LARCENY THEFT	NONE	GRAND THEFT FROM LOCKED AUTO
15	2015-05-13 22:00:00	200	WILLIAMS AV	BAYVIEW	-122.39774322509766	37.72993469238281	NON-CRIMINAL	NONE	FOUND PROPERTY
16	2015-05-13 22:00:00	0	MENDELL ST	BAYVIEW	-122.3836898803711	37.74319076538086	NON-CRIMINAL	NONE	FOUND PROPERTY
17	2015-05-13 22:00:00	EDDY ST	JONES ST	TENDERLOIN	-122.41259765625	37.783931732177734	ROBBERY	NONE	ROBBERY / ARMED WITH A KNIFE
18	2015-05-13 21:55:00	GODEUS ST	MISSION ST	INGLESIDE	-122.42168426513672	37.742820739746094	ASSAULT	NONE	AGGRAVATED ASSAULT WITH BODILY FORCE
19	2015-05-13 21:40:00	MENDELL ST	HUDSON AV	BAYVIEW	-122.38639831542969	37.738983154296875	OTHER OFFENSES	ARREST BOOKED	TRAFFIC VIOLATION
20	2015-05-13 21:30:00	100	JONES ST	TENDERLOIN	-122.41224670410156	37.782554626464844	NON-CRIMINAL	NONE	FOUND PROPERTY
21	2015-05-13 21:30:00	200	EVELYN WY	INGLESIDE	-122.44938659667969	37.74266815185547	LARCENY THEFT	NONE	GRAND THEFT FROM LOCKED AUTO
22	2015-05-13 21:17:00	1600	VALENCIA ST	INGLESIDE	-122.42027282714844	37.74733352661133	ROBBERY	NONE	ROBBERY / BODILY FORCE
23	2015-05-13 21:11:00	100	JONES ST	TENDERLOIN	-122.41224670410156	37.782554626464844	WARRANTS	NONE	WARRANT ARREST
24	2015-05-13 21:11:00	100	JONES ST	TENDERLOIN	-122.41224670410156	37.782554626464844	NON-CRIMINAL	NONE	STAY AWAY OR COURT ORDER / NON-DV RELATED
25	2015-05-13 21:10:00	FILLMORE ST	LOMBARD ST	NORTHERN	-122.43605041503906	37.799842834472656	LARCENY THEFT	NONE	GRAND THEFT FROM LOCKED AUTO

25 rows in set (0.00 sec)

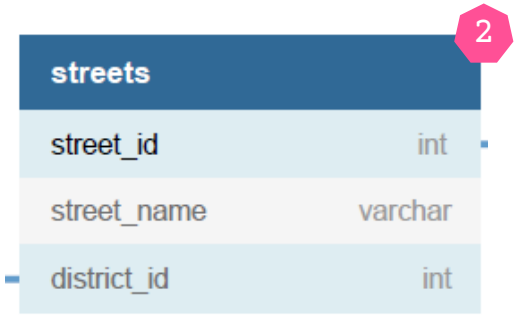
Data Ingestion: *'districts'* Table




```
INSERT INTO districts (district_name)
  SELECT DISTINCT district
  FROM dnorm_data;
```

```
mysql> SELECT * FROM districts;
+-------------+---------------+
| district_id | district_name |
+-------------+---------------+
|           4 | BAYVIEW       |
|           6 | CENTRAL       |
|           3 | INGLESIDE     |
|           9 | MISSION       |
|           1 | NORTHERN      |
|           2 | PARK          |
|           5 | RICHMOND      |
|          10 | SOUTHERN      |
|           7 | TARAVAL       |
|           8 | TENDERLOIN    |
+-------------+---------------+
10 rows in set (0.02 sec)
```

Data Ingestion: *'streets'* Table




```
INSERT INTO streets (street_name, district_id)
  SELECT s.street, d.district_id
  FROM ( SELECT DISTINCT street, district
         FROM dnorm_data
  ) s
  LEFT JOIN districts d
  ON s.district = d.district_name;
```

```
mysql> SELECT * FROM streets LIMIT 25;
+-----------+-------------+-------------+
| street_id | street_name | district_id |
+-----------+-------------+-------------+
|       324 | 10TH AV     |           7 |
|        42 | 10TH ST     |          10 |
|       201 | 11TH AV     |           5 |
|       168 | 11TH AV     |           7 |
|       276 | 11TH ST     |          10 |
|       231 | 14TH AV     |           5 |
|        34 | 14TH AV     |           7 |
|       282 | 14TH ST     |           2 |
|       215 | 14TH ST     |           9 |
|       268 | 15TH AV     |           5 |
|       291 | 15TH ST     |           9 |
|        69 | 16TH ST     |           2 |
|       263 | 16TH ST     |           9 |
|       251 | 17TH AV     |           5 |
|       192 | 17TH AV     |           7 |
|       395 | 17TH ST     |           2 |
|       158 | 17TH ST     |           9 |
|        52 | 18TH ST     |           2 |
|       200 | 18TH ST     |           9 |
|       359 | 19TH AV     |           7 |
|       334 | 19TH ST     |           9 |
|       313 | 1ST ST      |          10 |
|       253 | 20TH AV     |           5 |
|       228 | 20TH AV     |           7 |
|        68 | 20TH ST     |           4 |
+-----------+-------------+-------------+
25 rows in set (0.02 sec)
```

Data Ingestion: *'locations'* Table

3

locations	
location_id	int
spot	varchar
street_id	int
latitude	float
longitude	float

```
mysql> SELECT * FROM locations LIMIT 10;
+-------------+------+----------+----------+----------+
| location_id | spot | street_id | latitude | longitude |
+-------------+------+----------+----------+----------+
|           5 | 0    |         5 | -122.403 |   37.7134 |
|          10 | 0    |        10 | -122.488 |   37.7377 |
|          14 | 0    |        14 | -122.384 |   37.7432 |
|         166 | 0    |        19 | -122.406 |   37.7867 |
|          46 | 0    |        27 | -122.403 |   37.7873 |
|          35 | 0    |        30 | -122.432 |   37.7108 |
|          55 | 0    |        31 | -122.395 |   37.7941 |
|          38 | 0    |        32 | -122.406 |    37.786 |
|          41 | 0    |        33 | -122.424 |   37.7352 |
|          53 | 0    |        43 | -122.401 |   37.7907 |
+-------------+------+----------+----------+----------+
10 rows in set (0.02 sec)
```

```
INSERT INTO locations (spot, street_id, latitude, longitude)
  SELECT dt.spot, sd.street_id, dt.latitude, dt.longitude
  FROM ( SELECT DISTINCT spot, street, district, latitude, longitude
         FROM dnorm_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;
```

Data Ingestion: *'incidences'* Table

```sql
INSERT INTO incidences (incidence_id, timestamp, location_id, description)
  SELECT idt.row_num, idt.timestamp, lsd.location_id, idt.description
  FROM dnorm_data idt
  LEFT JOIN ( SELECT DISTINCT lc.location_id, lc.spot, 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.spot = lsd.spot
  AND idt.street = lsd.street_name
  AND idt.district = lsd.district_name;
```

4

incidences	
incidence_id	int
timestamp	datetime
location_id	int
description	varchar

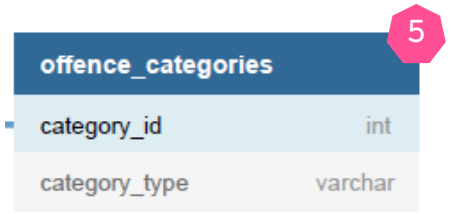
Data Ingestion: *'incidences'* Table

4

incidences	
incidence_id	int
timestamp	datetime
location_id	int
description	varchar

```
mysql> SELECT * FROM incidences LIMIT 25;
+--------------+---------------------+-------------+-----------------------------------------+
| incidence_id | timestamp           | location_id | description                             |
+--------------+---------------------+-------------+-----------------------------------------+
|            1 | 2015-05-13 23:53:00 |           1 | WARRANT ARREST                          |
|            2 | 2015-05-13 23:53:00 |           1 | TRAFFIC VIOLATION ARREST                |
|            3 | 2015-05-13 23:33:00 |           2 | TRAFFIC VIOLATION ARREST                |
|            4 | 2015-05-13 23:30:00 |           3 | GRAND THEFT FROM LOCKED AUTO            |
|            5 | 2015-05-13 23:30:00 |           4 | GRAND THEFT FROM LOCKED AUTO            |
|            6 | 2015-05-13 23:30:00 |           5 | GRAND THEFT FROM UNLOCKED AUTO          |
|            7 | 2015-05-13 23:30:00 |           6 | STOLEN AUTOMOBILE                       |
|            8 | 2015-05-13 23:30:00 |           7 | STOLEN AUTOMOBILE                       |
|            9 | 2015-05-13 23:00:00 |           8 | GRAND THEFT FROM LOCKED AUTO            |
|           10 | 2015-05-13 23:00:00 |           9 | GRAND THEFT FROM LOCKED AUTO            |
|           11 | 2015-05-13 22:58:00 |           9 | PETTY THEFT FROM LOCKED AUTO            |
|           12 | 2015-05-13 22:30:00 |          10 | MISCELLANEOUS INVESTIGATION             |
|           13 | 2015-05-13 22:30:00 |          11 | MALICIOUS MISCHIEF / VANDALISM OF VEHICLES |
|           14 | 2015-05-13 22:06:00 |          12 | GRAND THEFT FROM LOCKED AUTO            |
|           15 | 2015-05-13 22:00:00 |          13 | FOUND PROPERTY                          |
|           16 | 2015-05-13 22:00:00 |          14 | FOUND PROPERTY                          |
|           17 | 2015-05-13 22:00:00 |          15 | ROBBERY / ARMED WITH A KNIFE            |
|           18 | 2015-05-13 21:55:00 |          16 | AGGRAVATED ASSAULT WITH BODILY FORCE    |
|           19 | 2015-05-13 21:40:00 |          17 | TRAFFIC VIOLATION                       |
|           20 | 2015-05-13 21:30:00 |          18 | FOUND PROPERTY                          |
|           21 | 2015-05-13 21:30:00 |          19 | GRAND THEFT FROM LOCKED AUTO            |
|           22 | 2015-05-13 21:17:00 |          20 | ROBBERY / BODILY FORCE                  |
|           23 | 2015-05-13 21:11:00 |          18 | WARRANT ARREST                          |
|           24 | 2015-05-13 21:11:00 |          18 | STAY AWAY OR COURT ORDER / NON-DV RELATED |
|           25 | 2015-05-13 21:10:00 |          21 | GRAND THEFT FROM LOCKED AUTO            |
+--------------+---------------------+-------------+-----------------------------------------+
25 rows in set (0.02 sec)
```

Data Ingestion: *'offence_categories'* Table




```
INSERT INTO offence_categories (category_type)
  SELECT DISTINCT category
  FROM dnorm_data;
```

```
mysql> SELECT * FROM offence_categories;
+-------------+----------------------------+
| category_id | category_type              |
+-------------+----------------------------+
|          57 | ARSON                      |
|          39 | ASSAULT                    |
|          41 | BURGLARY                   |
|          56 | DISORDERLY CONDUCT         |
|          53 | DRIVING UNDER THE INFLUENCE |
|          45 | DRUG NARCOTIC              |
|          43 | DRUNKENNESS                |
|          58 | FAMILY OFFENSES            |
|          44 | FORGERY COUNTERFEITING     |
|          50 | FRAUD                      |
|          51 | KIDNAPPING                 |
|          34 | LARCENY THEFT              |
|          49 | MISSING PERSON             |
|          37 | NON-CRIMINAL               |
|          33 | OTHER OFFENSES             |
|          55 | PROSTITUTION               |
|          38 | ROBBERY                    |
|          52 | RUNAWAY                    |
|          47 | SECONDARY CODES            |
|          54 | SEX OFFENSES FORCIBLE      |
|          46 | STOLEN PROPERTY            |
|          42 | SUSPICIOUS OCC             |
|          48 | TRESPASS                   |
|          36 | VANDALISM                  |
|          35 | VEHICLE THEFT              |
|          32 | WARRANTS                   |
|          40 | WEAPON LAWS                |
+-------------+----------------------------+
27 rows in set (0.02 sec)
```

Data Ingestion: *'offences'* Table


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

```
mysql> SELECT * FROM offences LIMIT 25;
+--------------+-------------+
| incidence_id | category_id |
+--------------+-------------+
|            1 |          32 |
|           23 |          32 |
|           67 |          32 |
|           68 |          32 |
|           76 |          32 |
|           77 |          32 |
|          117 |          32 |
|          118 |          32 |
|          141 |          32 |
|          144 |          32 |
|          175 |          32 |
|          183 |          32 |
|          187 |          32 |
|          191 |          32 |
|          200 |          32 |
|          216 |          32 |
|          220 |          32 |
|          235 |          32 |
|          257 |          32 |
|          271 |          32 |
|          312 |          32 |
|          354 |          32 |
|          431 |          32 |
|          472 |          32 |
|          494 |          32 |
+--------------+-------------+
25 rows in set (0.00 sec)
```

Data Ingestion: *'resolution_types'* Table

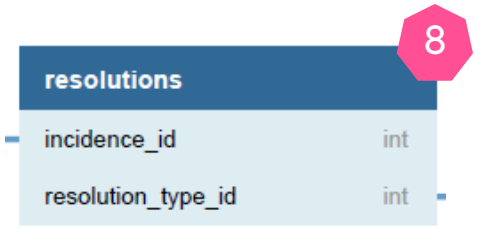
7

resolution_types	
resolution_type_id	int
resolution_type	varchar

```
INSERT INTO resolution_types (resolution_type)
  SELECT DISTINCT resolution
  FROM dnorm_data
  WHERE LOWER(resolution) NOT LIKE '%none%';
```

```
mysql> SELECT * FROM resolution_types;
+--------------------+-----------------------+
| resolution_type_id | resolution_type       |
+--------------------+-----------------------+
|                 29 | ARREST BOOKED         |
|                 30 | ARREST CITED          |
|                 34 | EXCEPTIONAL CLEARANCE |
|                 32 | JUVENILE BOOKED       |
|                 35 | LOCATED               |
|                 31 | PSYCHOPATHIC CASE     |
|                 33 | UNFOUNDED             |
+--------------------+-----------------------+
7 rows in set (0.01 sec)
```

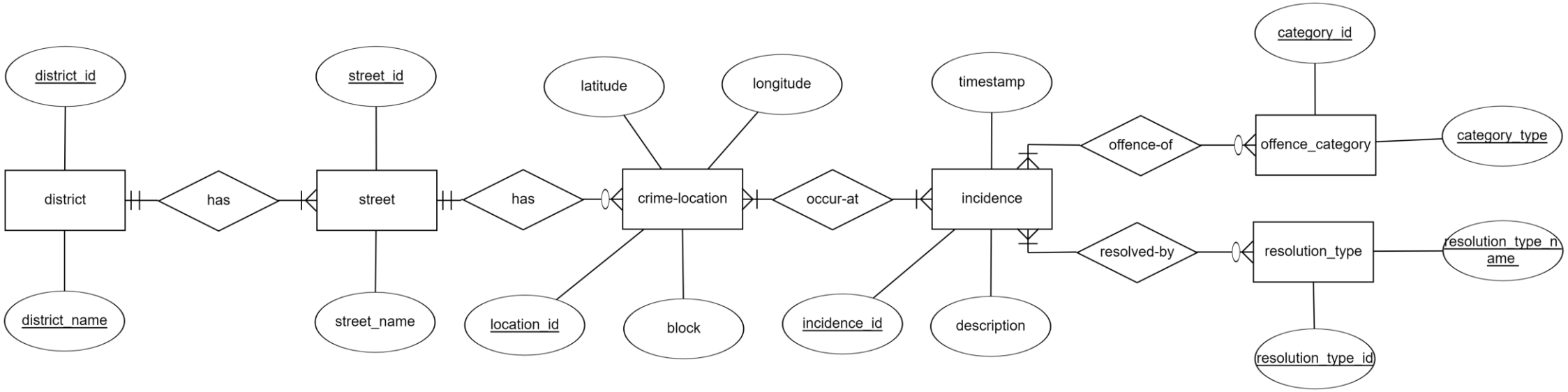
Data Ingestion: *'resolutions'* Table




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

```
mysql> SELECT * FROM resolutions LIMIT 25;
+--------------+--------------------+
| incidence_id | resolution_type_id |
+--------------+--------------------+
|            1 |                 29 |
|            2 |                 29 |
|            3 |                 29 |
|           19 |                 29 |
|           32 |                 29 |
|           44 |                 29 |
|           45 |                 29 |
|           50 |                 29 |
|           80 |                 29 |
|           85 |                 29 |
|           97 |                 29 |
|          103 |                 29 |
|          104 |                 29 |
|          105 |                 29 |
|          112 |                 29 |
|          114 |                 29 |
|          123 |                 29 |
|          126 |                 29 |
|          137 |                 29 |
|          138 |                 29 |
|          141 |                 29 |
|          144 |                 29 |
|          148 |                 29 |
|          161 |                 29 |
|          175 |                 29 |
+--------------+--------------------+
25 rows in set (0.01 sec)
```

Entity-Relationship (ER) Diagram



https://erdplus.com

Express: Web App Introduction

Step 1: Create new directories called ‘`mid-term`’ and ‘clf-crime-watch’, where ‘`clf-crime-watch`’ is a subdirectory of ‘`mid-term`’.

```
$ mkdir mid-term
$ mkdir mid-term/clf-crime-watch
```

Skip steps ‘1’ & ‘2’ if the folders have already been created.

Step 2: cd into the new directory.

```
$ cd mid-term/clf-crime-watch
```

Step 3: Inside `clf-crime-watch`, create a new file called ‘`app.js`’. the new directory.

```
$ touch app.js
```

Express: Web App Introduction (cont.)

Step 4: Initialise `npm` with `'app.js'` as start point by setting `package name` and `entry point` to `'clf-crime-watch'` and `'app.js'`, respectively.

```
$ npm init
```

Step 5: Install `'express'` to be used with our 'clif-crime-watch' node.js Web App.

```
$ npm install express
```

Step 6: `'express'` is now added as `'clf-crime-watch'` package dependencies. Open `'package.json'` in the editor and look for the following lines:

```
"dependencies": {
  "express": "^4.17.1"
}
```

Express: Web App Introduction (Cont.)

Step 7: Edit `'app.js'` file with the following content: (codes are in the shared `'web-app-intro.js'` file.)

```
JS app.js > ...
const express = require('express');
const app = express();
const port = 3000;

app.get('/', function (req, res) {
    res.send('<h1>Hello, World!</h1>');
})

app.listen(port, function () {
  console.log('The app is listening at http://localhost:${port}.')
})
```

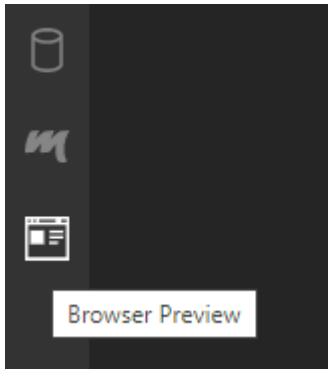
Express: Web App Introduction (cont.)

Step 8: Run our Web App using the following command: (to exit type Ctrl+C)

```
$ node app.js
```

Step 9: Open Coursera Lab's web browser, located on the *side* menu bar and enter the following URL:

```
localhost:3000
```



Step 10: We should be able to see a `H1` text `'Hello, World!'` being rendered to the browser.

Express + MySQL

Step 1: Install ‘`mysql`’ to be used with our ‘clif-crime-watch’ node.js Web App.

```
$ npm install mysql
$ npm install body-parser
$ npm install mustache-express
$ npm install yallist
```

Step 2: ‘`mysql`’, ‘`body-parser`’, ‘`mustache-express`’ and ‘`yallist`’ are now added as ‘`clf-crime-watch`’ package dependencies. Open ‘`package.json`’ in the editor and look for the following lines:

```
"dependencies": {
  "body-parser": "^1.19.0",
  "express": "^4.17.1",
  "mustache-express": "^1.3.2",
  "mysql": "^2.18.1",
  "yallist": "^4.0.0"
}
```

Express + MySQL (Cont.)

Step 3: Create a new directory called ‘templates’, where ‘templeates’ is a subdirectory of ‘clf_crime_watch’.

```
$ mkdir templates
```

Step 4: cd into the new directory.

```
$ cd templates
```

Step 5: Inside `templates`, create a new file called ‘`index.html`’. the new directory.

```
$ touch index.html
```

Express + MySQL (Cont.)

Step 6: Edit `'index.html'` file with the following content:

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8" />
    <title>Districts in San Francisco</title>
</head>

<body>
    <table>
        {{#data}}
        <tr>
            <td>{{district_id}}</td>
            <td>{{district_name}}</td>
        </tr>
        {{/data}}
    </table>
</body>

</html>
```

Express + MySQL (Cont.)

Step 7: Edit '`app.js`' file with the following content: (codes are in the shared '`express-mysql.js`' file.)

1

```javascript
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 }));

var dbcon = mysql.createConnection({
    host: 'localhost',
    user: 'francis',
    password: 'california',
    database: 'clf_crime_data'
})

```

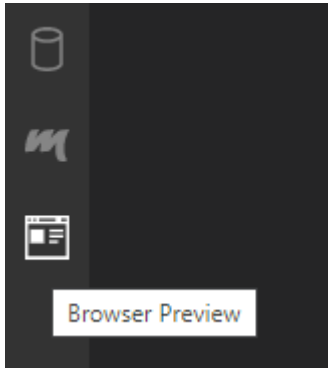
2

```javascript
function templateRenderer(template, res) {
    return function (error, results, fields) {
        if (error)
            throw error;

        res.render(template, { data: results });
    }
}

app.get('/', function (req, res) {
    dbcon.connect();
    db.query("SELECT * FROM districts;", templateRenderer('index', res));
    dbcon.end();
})

app.listen(port, function () {
    console.log('The app is listening at http://localhost:' + port + '.');
})

```

Express + MySQL (cont.)

Step 8: Run our Web App using the following command: (to exit type Ctrl+C)

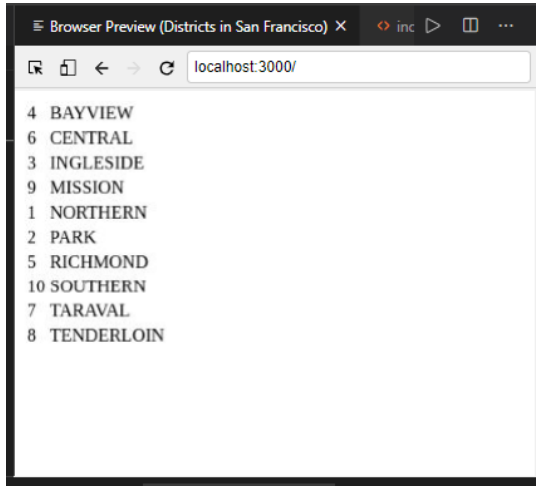
```
$ node app.js
```



Step 9: Open Coursera Lab's web browser, located on the *side* menu bar and enter the following URL:

```
localhost:3000
```

Step 10: We should be able to see a list of districts contained in our `'districts'` data table being rendered to the browser.



Appendix

Normalising a Table

NORMALIZING A TABLE

Princess Ruruna and Cain learned about normalization, the process of tabulating data from the real world for a relational database. It is necessary to normalize data in order to properly manage a relational database. Normalization is summarized here (the shaded fields are *primary keys*).

UNNORMALIZED FORM

Report code	Date	Export destination code	Export destination name	Product code	Product name	Unit price	Quantity

FIRST NORMAL FORM

Report code	Date	Export destination code	Export destination name

Report code	Product code	Product name	Unit price	Quantity

SECOND NORMAL FORM

Report code	Date	Export destination code	Export destination name

Report code	Product code	Quantity

Product code	Product name	Unit price

THIRD NORMAL FORM

Report code	Date	Export destination code

Export destination code	Export destination name

Report code	Product code	Quantity

Product code	Product name	Unit price

The *unnormalized form* is a table in which items that appear more than once have not been removed. We've seen that you cannot manage data well using this kind of table for a relational database. Consequently, you need to divide the table.

The *first normal form* refers to a simple, two-dimensional table resulting from division of the original, unnormalized table. You can consider it to be a table with one item in each cell. The table is divided so that no items will appear more than once.

The *second normal form* refers to a table in which a key that can identify data determines values in other columns. Here, it is the *primary key* that determines values in other columns.

In a relational database, a value is called *functionally dependent* if that value determines values in other columns. In the second normal form, the table is divided so that values in other columns are functionally dependent on the primary key.

In the *third normal form*, a table is divided so that a value is not determined by any non-primary key. In a relational database, a value is called *transitively dependent* if that value determines values in other columns indirectly, which is part of functionally dependent operation. In the third normal form, the table is divided so that transitively dependent values are removed.

Normalising a Table: Examples

Q8

The following table represents an order-receiving system. Normalize it to the third normal form. However, process one customer per order-taking code. You can process multiple products based on one order-taking code. In addition, one order-taking code should correspond to only one representative.

Order-taking code	Date	Customer code	Customer name	Product code	Product name	Unit price	Represen-tative code	Represen-tative name	Quantity

Order-taking code	Date	Customer code	Representative code

Customer code	Customer name

Order-taking code	Product code	Quantity

Product code	Product name	Unit price

Representative code	Representative name

Q9

The following table represents an order-receiving system. Normalize it to the third normal form. Assume that products are classified by product code.

Order-taking code	Date	Customer code	Customer name	Product code	Product name	Unit price	Product classification code	Product classification name	Quantity

Order-taking code	Date	Customer code

Customer code	Customer name

Order-taking code	Product code	Quantity

Product code	Product classification code	Product name	Unit price

Product classification code	Product classification name