capstone_project_part1

July 24, 2021

Modeling and Forecasting Crime Rate in Colorado

Data Science Capstone Project, part I * Student name: Elena Kazakova * Student pace: Full-time * Cohort: DS02222021 * Scheduled project review date: 07/26/2021 * Instructor name: James Irving * Application url: TBD

TABLE OF CONTENTS

This notebook is Part I of the project. Its' goal is to pre-process data in the SQLite database in order to use it for building DataFrames in the modeling part of the project. Part ZERO is in the notebook dedicated to creating a SQLite database, uploading and partially cleaning the tables. The link to the part ZERO notebook.

1 IMPORTS

If you are running this notebook without restarting the kernel replace '%load_ext autoreload' in imports with '%reload_ext autoreload'

[1]:

```
# Importing packages
import pandas as pd
from pandasql import sqldf
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pickle
import gzip
import shutil
import os
import sqlite3
import db_to_sqlite
from sqlite3 import Error
import csv
from pathlib import Path
import subprocess
import io
from icecream import ic
import warnings
warnings.filterwarnings(action='ignore', category=FutureWarning)
from functions_all import *

%load_ext autoreload
%autoreload 2
%matplotlib inline
```

2 OBTAIN

2.1 Data

2.1.1 Data source and data description

The data description part is duplicated in the part 0 notebook

Data is from FBI Crime Data Explorer NIBRS data for Colorado from 2009-2019

The data dictionary is and a record descriptiopn are available.

The description of the main and reference tables is in data/README.md file. The agency implemented some changes to the files structure in 2016 and removed the sqlite create and load scripts from the zip directories. Another fact worth mentioning is that files 'nibrs_property_desc.csv' from 2014 and 2015 have duplicated nibrs_property_desc_ids (unique identifier in the nibrs_property_desc table) which complicated the loading of the data.

All 2016-2019 files need to be cleaned up because FBI changed the file format. There is a YEAR column that needs to be removed as well as the legacy columns from the previous years need to be added up. It's a tedious job and it needs to be done once and the files need to be backed up.

In order to clean the tables up the following needs to be done

1. Remove all **DATA_YEAR** columns from each file, it's the first column

2. Files that do not need any changes beyond **DATA_YEAR** column removal

   nibrs_arrestee_weapon.csv nibrs_bias_motivation.csv nibrs_criminal_act.csv nibrs_property_desc.csv nibrs_suspect_using.csv nibrs_suspected_drug.csv nibrs_victim_circumstances.csv nibrs_victim_injury.csv nibrs_victim_offender_rel.csv nibrs_victim_offense.csv nibrs_weapon.csv

3. in **nibrs_arestee.csv file**:

a. between **ARRESTEE_SEQ_NUM** and **ARREST_DATE** there should be an **arrest_num column**
b. Between **CLEARANCE_IND** and **AGE_RANGE_LOW_NUM** should be a **ff_line_number** column.

4. in **nibrs_incident** file: a.between **NIBRS_MONTH_ID** and **CARGO_THEFT_FLAG** column **incident_number** b.between **DATA_HOME** and **ORIG_FORMAT** column **ddocname** c.between **ORIG_FORMAT** and **DID** column **ff_line_number**

5. in **nibrs_month.csv** file: a.between **REPORT_DATE** and **UPDATE_FLAG** add **prepared_date** column b.between **ORIG_FORMAT** and **DATA_HOME** column **ff_line_number** c.column **MONTH_PUB_STATUS** removed

6. in **nibrs_offender.csv** file: a.between **ETHNICITY_ID** and **AGE_RANGE_LOW_NUM** column **ff_line_number**

7. in **nibrs_offense.csv** file:

   a. the last column **ff_line_number** should be added

8. in **nibrs_property.csv** file:

   a. the last column **ff_line_number** should be added

9. in **nibrs_victim.csv** file:

   a. between **RESIDENT_STATUS_CODE** and **AGE_RANGE_LOW_NUM** two columns **agency_data_year** and **ff_line_number** (in that order) should be added

2.1.2 Using an already created sqlite database

The notebook with database creation is here. The referenced database is in ***data/sqlite/db/production1 db***. It takes 2.5 minutes to run the database creation code in the notebook.

[3]:

```
# Uncomment the line below if you are re-running the code part for main tables OR if you want to re-run all of the code
# withought re-running the database creating notebook>>> Run the first command only if you want to re-use production1
# database and comment it out if you re-ran the create database notebook just before switching to this one.
```

```
!cp data/sqlite/db/production1_backup.db data/sqlite/db/production1.db

!cp data/sqlite/db/production1.db data/sqlite/db/production1_backup.db
```

[4]:
```
# Initiating a cursor
conn = sqlite3.connect('data/sqlite/db/production1.db')
cur = conn.cursor()
```

[5]:
```
q="""SELECT name FROM sqlite_master WHERE type='table'"""
df=table_query(q, cur)
df
```

[5]:
```
                         name
0                    agencies
1        agency_participation
2                cde_agencies
3         nibrs_activity_type
4                   nibrs_age
5           nibrs_arrest_type
6       nibrs_assignment_type
7             nibrs_bias_list
8         nibrs_location_type
9          nibrs_offense_type
10       nibrs_prop_desc_type
11          nibrs_victim_type
12        nibrs_circumstances
13       nibrs_cleared_except
14         nibrs_criminal_act
15    nibrs_criminal_act_type
16    nibrs_drug_measure_type
17            nibrs_ethnicity
18               nibrs_injury
19    nibrs_justifiable_force
20       nibrs_prop_loss_type
21         nibrs_relationship
22  nibrs_suspected_drug_type
23           nibrs_using_list
24          nibrs_weapon_type
25                   ref_race
26                  ref_state
27             nibrs_arrestee
28      nibrs_arrestee_weapon
29      nibrs_bias_motivation
30                nibrs_month
31             nibrs_incident
32             nibrs_offender
33              nibrs_offense
```

```
34              nibrs_property
35         nibrs_property_desc
36         nibrs_suspect_using
37        nibrs_suspected_drug
38                nibrs_victim
39  nibrs_victim_circumstances
40         nibrs_victim_injury
41   nibrs_victim_offender_rel
42        nibrs_victim_offense
43                nibrs_weapon
```

[6]:

```
q="SELECT * FROM nibrs_incident"
df=table_query(q, cur)
df
```

[6]:

```
         agency_id  incident_id  nibrs_month_id incident_number  \
0             1971     51264520         4814762        09000019
1             1971     51264521         4814762        09000053
2             1971     51264523         4814762        09000082
3             1971     51264524         4814762        09000092
4             1971     51264525         4814762        09000097
…            …          …             …             …
2819458       2023    120337425         8226741
2819459       2023    119323671         8226741
2819460       2023    119323654         8226741
2819461       2023    120333220         8211417
2819462       2023    120337420         8219079

        cargo_theft_flag submission_date        incident_date  \
0                                         2009-01-05 00:00:00
1                                         2009-01-13 00:00:00
2                                         2009-01-17 00:00:00
3                                         2009-01-20 00:00:00
4                                         2009-01-21 00:00:00
…                  …             …                  …
2819458                N       11-Feb-20            17-Dec-19
2819459                        13-Jan-20            21-Dec-19
2819460                        13-Jan-20            19-Dec-19
2819461                        11-Feb-20            13-Oct-19
2819462                N       11-Feb-20            24-Nov-19

        report_date_flag incident_hour  cleared_except_id cleared_except_date  \
0                                   22                  6
1                                                       6
2                                   19                  6
3                      R                                6
4                                                       6
```

```
…                  …           …                …                 …
2819458                               9                   6
2819459                              14                   6
2819460                              22                   6
2819461                              13                   6
2819462                              13                   6

         incident_status data_home                             ddocname  \
0                      0         C  2009_01_CO0320000_09000019_INC_NIBRS
1                      0         C  2009_01_CO0320000_09000053_INC_NIBRS
2                      0         C  2009_01_CO0320000_09000082_INC_NIBRS
3                      0         C  2009_01_CO0320000_09000092_INC_NIBRS
4                      0         C  2009_01_CO0320000_09000097_INC_NIBRS
…                  …       …                                    …
2819458                0         C
2819459                0         C
2819460                0         C
2819461                0         C
2819462                0         C

        orig_format ff_line_number       did
0
1
2
3
4
…            …            …       …
2819458           F                 65195613
2819459           F                 63283836
2819460           F                 63283811
2819461           F                 65196826
2819462           F                 65196843

[2819463 rows x 17 columns]
```

[7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2819463 entries, 0 to 2819462
Data columns (total 17 columns):
 #   Column               Dtype
---  ------               -----
 0   agency_id            int64
 1   incident_id          int64
 2   nibrs_month_id       int64
 3   incident_number      object
 4   cargo_theft_flag     object
 5   submission_date      object
 6   incident_date        object
 7   report_date_flag     object
 8   incident_hour        object
 9   cleared_except_id    int64
 10  cleared_except_date  object
 11  incident_status      int64
 12  data_home            object
 13  ddocname             object
 14  orig_format          object
 15  ff_line_number       object
 16  did                  object
dtypes: int64(5), object(12)
memory usage: 365.7+ MB
```

3 SCRUB, part 1

3.1 SQL/cleaning tables

3.1.1 Main tables

[8]:
```
# df at this point is the main incident table, I am displaying it's info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2819463 entries, 0 to 2819462
Data columns (total 17 columns):
 #   Column               Dtype
---  ------               -----
 0   agency_id            int64
 1   incident_id          int64
 2   nibrs_month_id       int64
 3   incident_number      object
 4   cargo_theft_flag     object
 5   submission_date      object
 6   incident_date        object
 7   report_date_flag     object
 8   incident_hour        object
 9   cleared_except_id    int64
 10  cleared_except_date  object
 11  incident_status      int64
 12  data_home            object
 13  ddocname             object
 14  orig_format          object
 15  ff_line_number       object
 16  did                  object
dtypes: int64(5), object(12)
memory usage: 365.7+ MB
```

Dropping unneeded tables

[9]:

```
#Dropping the tables irrelavant to modeling and the dashboard

table_list_to_drop=['nibrs_month','nibrs_justifiable_force','nibrs_arrest_type',
                   ␣
 ↪'nibrs_drug_measure_type','nibrs_injury','nibrs_suspect_using',
                   ␣
 ↪'nibrs_suspected_drug','nibrs_suspected_drug_type','nibrs_using_list','nibrs_arrestee',
                   ␣
 ↪'nibrs_arrestee_weapon','nibrs_activity_type','nibrs_assignment_type','nibrs_property',
                   ␣
 ↪'nibrs_property_desc','nibrs_prop_loss_type','nibrs_victim_injury','nibrs_prop_desc_type',
                   ␣
 ↪'nibrs_circumstances','nibrs_victim_circumstances','ref_state',␣
 ↪'nibrs_criminal_act',
                    'nibrs_criminal_act_type','nibrs_victim_offense']

for table in table_list_to_drop:
    string=table
    statement='DROP TABLE'+' '+string
    cur.execute(statement)


cur.execute("""SELECT name FROM sqlite_master WHERE type='table'""").fetchall()
```

[9]:

```
[('agencies',),
 ('agency_participation',),
 ('cde_agencies',),
 ('nibrs_age',),
 ('nibrs_bias_list',),
 ('nibrs_location_type',),
 ('nibrs_offense_type',),
 ('nibrs_victim_type',),
 ('nibrs_cleared_except',),
 ('nibrs_ethnicity',),
 ('nibrs_relationship',),
 ('nibrs_weapon_type',),
 ('ref_race',),
 ('nibrs_bias_motivation',),
 ('nibrs_incident',),
 ('nibrs_offender',),
 ('nibrs_offense',),
 ('nibrs_victim',),
 ('nibrs_victim_offender_rel',),
 ('nibrs_weapon',)]
```

Incidents table

[10]:
```
#Listing columns in the incidents table

df.columns
```

[10]:
```
Index(['agency_id', 'incident_id', 'nibrs_month_id', 'incident_number',
       'cargo_theft_flag', 'submission_date', 'incident_date',
       'report_date_flag', 'incident_hour', 'cleared_except_id',
       'cleared_except_date', 'incident_status', 'data_home', 'ddocname',
       'orig_format', 'ff_line_number', 'did'],
      dtype='object')
```

[11]:
```
# statement1='DROP TABLE incident_main'
# cur.execute(statement1)
```

[12]:
```
# Creating a list of columns to leave in the incidents table

incdnt_clmns_to_lv=['agency_id','incident_id','incident_date','incident_hour']

# Due to the fact that sqlite has a limitation of not being able to drop␣
 ↪columns,
# I need to create a new table with only the columns I need.

create_new_table('nibrs_incident', 'incident_main', incdnt_clmns_to_lv, cur)
```

[12]:
```
         agency_id  incident_id        incident_date incident_hour
0             1971     51264520  2009-01-05 00:00:00            22
1             1971     51264521  2009-01-13 00:00:00
2             1971     51264523  2009-01-17 00:00:00            19
3             1971     51264524  2009-01-20 00:00:00
4             1971     51264525  2009-01-21 00:00:00
…              …           …                    …            …
2819458       2023    120337425            17-Dec-19             9
2819459       2023    119323671            21-Dec-19            14
2819460       2023    119323654            19-Dec-19            22
2819461       2023    120333220            13-Oct-19            13
2819462       2023    120337420            24-Nov-19            13

[2819463 rows x 4 columns]
```

Offense table

[13]:
```
# Main offense table columns

q='SELECT * FROM nibrs_offense'
df=table_query(q,cur)
df.head()
```

[13]:
```
   offense_id  incident_id  offense_type_id attempt_complete_flag  \
0    53563151     51264520               27                     C
1    53563402     51264521               14                     C
2    53558278     51264523               16                     C
3    53558279     51264523               35                     C
4    53563403     51264524               46                     C

   location_id num_premises_entered method_entry_code ff_line_number
0           20
1           20
2           22
3           22
4           25
```

[14]:
```
# Creating a list with columns to leave in the main offense table

offns_clmns_to_lv=['offense_id','incident_id','offense_type_id', 'location_id']

# Due to the fact that sqlite has a limitation of not being able to drop␣
 ↪columns,
# I need to create a new table with only the columns I need.

create_new_table('nibrs_offense', 'offense_main', offns_clmns_to_lv, cur)
```

[14]:
```
         offense_id  incident_id  offense_type_id  location_id
0          53563151     51264520               27           20
1          53563402     51264521               14           20
2          53558278     51264523               16           22
3          53558279     51264523               35           22
4          53563403     51264524               46           25
…             …          …              …          …
3201138   141844716    116813642                5           18
3201139   141852632    116813645               35            8
3201140   141848922    116813645               16            8
3201141   141844745    116813666               16           38
3201142   141848949    116813669               49           20

[3201143 rows x 4 columns]
```

Offender table

[15]:
```
# Main offender table columns

q='SELECT * FROM nibrs_offender'
df=table_query(q, cur)
df.columns
```

```
[15]: Index(['offender_id', 'incident_id', 'offender_seq_num', 'age_id', 'age_num',
             'sex_code', 'race_id', 'ethnicity_id', 'ff_line_number',
             'age_range_low_num', 'age_range_high_num'],
            dtype='object')
```

```
[16]: # Creating a list with columns to leave in the main offender table

      offndr_clmns_to_lv=['offender_id', 'incident_id','age_id',␣
       ↪'age_num','sex_code', 'race_id', 'ethnicity_id']

      # Due to the fact that sqlite has a limitation of not being able to drop␣
       ↪columns,
      # I need to create a new table with only the columns I need.

      create_new_table('nibrs_offender', 'offender_main', offndr_clmns_to_lv, cur)
```

```
[16]:          offender_id  incident_id age_id age_num sex_code race_id ethnicity_id
      0           57702592     51264520      5      25        M       1
      1           57702593     51264521
      2           57702595     51264523      5      20        M       1
      3           57702596     51264524
      4           57702597     51264525      5      55        M       1
      …                …            …     …      …       …      …           …
      3197986    133662374    117658122      5      35        M       1            2
      3197987    133662375    117658122      5      24        M       1            2
      3197988    133652539    117658122      5      30        M       1            2
      3197989    133662412    117658140      5      30        M       1            1
      3197990    133652562    117658144      5      12        M       1            2

      [3197991 rows x 7 columns]
```

```
[17]: # Using reference table values in the offender_main table. Replacing codes with␣
       ↪values comprehensible to humans.
      # I am doing it to simplify creating a dashboard later.

      df=add_update_clmn('offender_main','ref_race', 'race', 'race_desc', 'race_id',␣
       ↪cur)

      df=add_update_clmn('offender_main','nibrs_age', 'age_group', 'age_name',␣
       ↪'age_id', cur)

      df=add_update_clmn('offender_main','nibrs_ethnicity', 'ethnicity',␣
       ↪'ethnicity_name', 'ethnicity_id', cur)

      df=update_value('offender_main', 'sex_code', "'F'", "'Female'", cur)

      df=update_value('offender_main', 'sex_code', "'M'", "'Male'", cur)
```

```
df=update_value('offender_main', 'sex_code', "'U'", "'Unknown'", cur)

q='SELECT * FROM offender_main'
df=table_query(q,cur)
df.head()
```

[17]:

```
   offender_id  incident_id age_id age_num sex_code race_id ethnicity_id  \
0     57702592     51264520      5      25     Male       1
1     57702593     51264521
2     57702595     51264523      5      20     Male       1
3     57702596     51264524
4     57702597     51264525      5      55     Male       1

    race     age_group ethnicity
0  White  Age in Years      None
1   None          None      None
2  White  Age in Years      None
3   None          None      None
4  White  Age in Years      None
```

[18]:

```
df.columns
```

[18]:

```
Index(['offender_id', 'incident_id', 'age_id', 'age_num', 'sex_code',
       'race_id', 'ethnicity_id', 'race', 'age_group', 'ethnicity'],
      dtype='object')
```

[19]:

```
# Creating a list with columns to leave in the main offender table. I am␣
 ↪dropping all obsolete old columns

ofndr_clmns_to_lv=['offender_id', 'incident_id', 'age_num', 'sex_code',
       'race', 'age_group', 'ethnicity']

# Due to the fact that sqlite has a limitation of not being able to drop␣
 ↪columns,
# I need to create a new table with only the columns I need, drop the old table␣
 ↪and rename the new one.

create_new_table('offender_main', 'offender_main_tmp', ofndr_clmns_to_lv, cur,␣
 ↪drop_rename=True)
```

[19]:

```
         offender_id  incident_id age_num sex_code    race     age_group  \
0           57702592     51264520      25     Male  White  Age in Years
1           57702593     51264521                    None          None
2           57702595     51264523      20     Male  White  Age in Years
3           57702596     51264524                    None          None
4           57702597     51264525      55     Male  White  Age in Years
```

```
…              …          …     …      …    …           …
3197986    133662374    117658122      35      Male  White  Age in Years
3197987    133662375    117658122      24      Male  White  Age in Years
3197988    133652539    117658122      30      Male  White  Age in Years
3197989    133662412    117658140      30      Male  White  Age in Years
3197990    133652562    117658144      12      Male  White  Age in Years

                      ethnicity
0                          None
1                          None
2                          None
3                          None
4                          None
…                          …
3197986  Not Hispanic or Latino
3197987  Not Hispanic or Latino
3197988  Not Hispanic or Latino
3197989      Hispanic or Latino
3197990  Not Hispanic or Latino

[3197991 rows x 7 columns]
```

Victim table

[20]:

```
# Main victim table columns

q='SELECT * FROM nibrs_victim'
df=table_query(q, cur)
df.columns
```

[20]:

```
Index(['victim_id', 'incident_id', 'victim_seq_num', 'victim_type_id',
       'assignment_type_id', 'activity_type_id', 'outside_agency_id', 'age_id',
       'age_num', 'sex_code', 'race_id', 'ethnicity_id',
       'resident_status_code', 'agency_data_year', 'ff_line_number',
       'age_range_low_num', 'age_range_high_num'],
      dtype='object')
```

[21]:

```
# Creating a list with columns to leave in the main victim table

vctm_clmns_to_lv=['victim_id', 'incident_id', 'victim_type_id',
                  'age_id','age_num', 'sex_code', 'race_id',
                  'ethnicity_id','resident_status_code']

# Due to the fact that sqlite has a limitation of not being able to drop␣
 ↪columns,
# I need to create a new table with only the columns I need.

create_new_table('nibrs_victim', 'victim_main', vctm_clmns_to_lv, cur)
```

[21]:

```
         victim_id  incident_id  victim_type_id age_id age_num sex_code  \
0          55514644     51264520              5      5      23        M
1          55514645     51264521              4      5      49        F
2          55514647     51264523              8
3          55514648     51264524              4      5      28        F
4          55514649     51264525              4      5      16        M
…            …          …             …    …     …      …
3229635  130091066    118751536              4      5      40        F
3229636  130095316    118751542              4      5      31        F
3229637  130095315    118751542              4      5      33        M
3229638  130091076    118742446              4      5      19        F
3229639  130085633    118751549              4      5      37        M

        race_id ethnicity_id resident_status_code
0             1            2                    R
1             1            3                    N
2
3             1            3                    R
4             1            3                    R
…         …          …                  …
3229635       8            2                    R
3229636       1            2                    N
3229637       1            2                    N
3229638       1            3                    R
3229639       1            2                    R

[3229640 rows x 9 columns]
```

[22]:

```
# Using reference table values in the victim_main table. Replacing codes with values comprehensible to humans.
# I am doing it to simplify creating a dashboard later

df=add_update_clmn('victim_main','ref_race', 'race', 'race_desc', 'race_id', cur)

df=add_update_clmn('victim_main','nibrs_age', 'age_group', 'age_name', 'age_id', cur)

df=add_update_clmn('victim_main','nibrs_ethnicity', 'ethnicity', 'ethnicity_name', 'ethnicity_id', cur)

df=add_update_clmn('victim_main','nibrs_victim_type', 'victim_type', 'victim_type_name', 'victim_type_id', cur)

df=update_value('victim_main', 'sex_code', "'F'", "'Female'", cur)

df=update_value('victim_main', 'sex_code', "'M'", "'Male'", cur)
```

```
df=update_value('victim_main', 'sex_code', "'U'", "'Unknown'", cur)

df=update_value('victim_main', 'resident_status_code', "'R'", "'Resident'", cur)

df=update_value('victim_main', 'resident_status_code', "'N'", "'Non-resident'",␣
 ↪cur)

df=df=update_value('victim_main', 'resident_status_code', "'U'", "'Unknown'",␣
 ↪cur)

q='SELECT * FROM victim_main'
df=table_query(q, cur)
df.head()
```

[22]:

```
   victim_id  incident_id  victim_type_id age_id age_num sex_code race_id  \
0   55514644     51264520               5      5      23     Male       1
1   55514645     51264521               4      5      49   Female       1
2   55514647     51264523               8
3   55514648     51264524               4      5      28   Female       1
4   55514649     51264525               4      5      16     Male       1

  ethnicity_id resident_status_code   race     age_group  \
0            2             Resident  White  Age in Years
1            3         Non-resident  White  Age in Years
2                                      None          None
3            3             Resident  White  Age in Years
4            3             Resident  White  Age in Years

                ethnicity              victim_type
0  Not Hispanic or Latino  Law Enforcement Officer
1                 Unknown               Individual
2                    None           Society/Public
3                 Unknown               Individual
4                 Unknown               Individual
```

[23]:

```
df.columns
```

[23]:

```
Index(['victim_id', 'incident_id', 'victim_type_id', 'age_id', 'age_num',
       'sex_code', 'race_id', 'ethnicity_id', 'resident_status_code', 'race',
       'age_group', 'ethnicity', 'victim_type'],
      dtype='object')
```

[24]:

```
# Creating a list with columns to leave in the main victim table. I am dropping␣
 ↪all obsolete old columns.

vctm_clmns_to_lv=['victim_id', 'incident_id', 'age_num',
          'sex_code', 'resident_status_code', 'race',
          'age_group', 'ethnicity', 'victim_type']

# Due to the fact that sqlite has a limitation of not being able to drop␣
 ↪columns,
# I need to create a new table with only the columns I need, drop the old table␣
 ↪and rename the new one.

create_new_table('victim_main', 'victim_main_tmp', vctm_clmns_to_lv, cur,␣
 ↪drop_rename=True)
```

[24]:

```
         victim_id  incident_id age_num sex_code resident_status_code  \
0         55514644     51264520      23     Male             Resident
1         55514645     51264521      49   Female         Non-resident
2         55514647     51264523
3         55514648     51264524      28   Female             Resident
4         55514649     51264525      16     Male             Resident
…            …          …     …      …                  …
3229635  130091066    118751536      40   Female             Resident
3229636  130095316    118751542      31   Female         Non-resident
3229637  130095315    118751542      33     Male         Non-resident
3229638  130091076    118742446      19   Female             Resident
3229639  130085633    118751549      37     Male             Resident

                                              race     age_group  \
0                                            White  Age in Years
1                                            White  Age in Years
2                                             None          None
3                                            White  Age in Years
4                                            White  Age in Years
…                                            …           …
3229635  Native Hawaiian or Other Pacific Islander  Age in Years
3229636                                      White  Age in Years
3229637                                      White  Age in Years
3229638                                      White  Age in Years
3229639                                      White  Age in Years

                      ethnicity              victim_type
0        Not Hispanic or Latino  Law Enforcement Officer
1                       Unknown               Individual
2                          None           Society/Public
3                       Unknown               Individual
4                       Unknown               Individual
…                         …                       …
3229635  Not Hispanic or Latino               Individual
3229636  Not Hispanic or Latino               Individual
3229637  Not Hispanic or Latino               Individual
```

```
3229638                Unknown               Individual
3229639  Not Hispanic or Latino               Individual

[3229640 rows x 9 columns]
```

Weapon table

[25]:
```
# Main weapon table columns

q='SELECT * FROM nibrs_weapon'
df=table_query(q, cur)
df.columns
```

[25]:
```
Index(['weapon_id', 'offense_id', 'nibrs_weapon_id'], dtype='object')
```

[26]:
```
# Creating a list with columns to leave in the main weapon table

wpn_clmns_to_lv=['weapon_id', 'offense_id']

# Due to the fact that sqlite has a limitation of not being able to drop␣
 ↪columns,
# I need to create a new table with only the columns I need.

create_new_table('nibrs_weapon', 'weapon_main', wpn_clmns_to_lv, cur)
```

[26]:
```
        weapon_id  offense_id
0              12    53563151
1              12    53558280
2              12    53563153
3              12    53579810
4              12    53572975
…             …          …
551044         12   138305073
551045          3   138310667
551046         12   141818270
551047         12   141833579
551048          3   141833723

[551049 rows x 2 columns]
```

[27]:
```
cur.execute("""SELECT name FROM sqlite_master WHERE type='table'""").fetchall()
```

[27]:
```
[('agencies',),
 ('agency_participation',),
 ('cde_agencies',),
 ('nibrs_age',),
 ('nibrs_bias_list',),
 ('nibrs_location_type',),
 ('nibrs_offense_type',),
 ('nibrs_victim_type',),
 ('nibrs_cleared_except',),
 ('nibrs_ethnicity',),
 ('nibrs_relationship',),
 ('nibrs_weapon_type',),
 ('ref_race',),
 ('nibrs_bias_motivation',),
 ('nibrs_incident',),
 ('nibrs_offender',),
 ('nibrs_offense',),
 ('nibrs_victim',),
 ('nibrs_victim_offender_rel',),
 ('nibrs_weapon',),
 ('incident_main',),
 ('offense_main',),
 ('offender_main',),
 ('victim_main',),
 ('weapon_main',)]
```

[28]:

```
q='SELECT * FROM weapon_main'
df=table_query(q, cur)
df.count()
```

[28]:

```
weapon_id     551049
offense_id    551049
dtype: int64
```

[29]:

```
q='SELECT * FROM nibrs_weapon_type'
df=table_query(q, cur)
df
```

[29]:

```
    weapon_id weapon_code                      weapon_name shr_flag
0          21         11A             Firearm (Automatic)        N
1          22         12A             Handgun (Automatic)        N
2          23         13A               Rifle (Automatic)        N
3          24         14A             Shotgun (Automatic)        N
4          25         15A       Other Firearm (Automatic)        N
5          26          55      Pushed or Thrown Out Window        Y
6          27          75                         Drowning        Y
7          28          80  Strangulation - Include Hanging        Y
8           1          01                          Unarmed        N
9           2          11                          Firearm        Y
10          3          12                          Handgun        Y
11          4          13                            Rifle        Y
12          5          14                          Shotgun        Y
13          6          15                    Other Firearm        Y
```

```
14          7          16        Lethal Cutting Instrument        N
15          8          17     Club/Blackjack/Brass Knuckles       N
16          9          20       Knife/Cutting Instrument          Y
17         10          30                    Blunt Object         Y
18         11          35                   Motor Vehicle         N
19         12          40                Personal Weapons         Y
20         13          50                          Poison         Y
21         14          60                      Explosives         Y
22         15          65          Fire/Incendiary Device         Y
23         16          70    Drugs/Narcotics/Sleeping Pills       Y
24         17          85                     Asphyxiation        Y
25         18          90                           Other         Y
26         19          95                         Unknown         N
27         20          99                            None         N
```

[30]:
```
# Intermediatly (to be dropped later) adding 'weapon_name' column to␣
 ↪weapon_main table, plus 'weapon' column
add_update_clmn('weapon_main','nibrs_weapon_type', 'weapon_name',␣
 ↪'weapon_name', 'weapon_id', cur)
cur.execute('ALTER TABLE weapon_main ADD COLUMN weapon')

# Making sure the columns are there
q='SELECT * FROM weapon_main'
df=table_query(q, cur)
df.head()
```

[30]:
```
   weapon_id  offense_id       weapon_name weapon
0         12    53563151  Personal Weapons   None
1         12    53558280  Personal Weapons   None
2         12    53563153  Personal Weapons   None
3         12    53579810  Personal Weapons   None
4         12    53572975  Personal Weapons   None
```

[31]:
```
# A snippet to change weapon_main by adding a weapon_name and a weapon columns␣
 ↪based on nibrs_weapon_type table values
# the final weapont_main will have only 2 columns offense_id and weapon with 5␣
 ↪unique values 'Unarmed','Unknown',
# 'Other weapon','Non-automatic firearm','Automatic firearm'.

# Anything with 'automatic' is mapped to 'Automatic firearm'
# 'Unknown' - to 'Unknown'
# 'Unarned'  or 'None' - to 'Unarmed'
# 'Firarm', 'Handgun','Rifle','Shotgun','Personal Weapons' or 'Other Firearm'␣
 ↪to 'Non-automatic firearm'
# the rest of values are mapped to 'Other weapon'
# I could've possibly done it by creating a dataframe, using dictionary to␣
 ↪update the values
# and kicking it back to the database.

statement="UPDATE weapon_main SET weapon='Automatic firearm' WHERE weapon_name␣
 ↪like ('%Automatic%')"
cur.execute(statement)

statement="UPDATE weapon_main SET weapon=weapon_name WHERE␣
 ↪weapon_name='Unknown'"
cur.execute(statement)

statement="UPDATE weapon_main SET weapon='Unarmed' WHERE weapon_name in␣
 ↪('None','Unarmed')"
cur.execute(statement)

statement="UPDATE weapon_main SET weapon='Non-automatic firearm' \
WHERE weapon_name in ('Firarm', 'Handgun','Rifle','Shotgun','Personal␣
 ↪Weapons','Other Firearm')"
cur.execute(statement)

statement="UPDATE weapon_main SET weapon='Other weapon' WHERE weapon is Null"
cur.execute(statement)

# Creating a list with columns to leave in the main weapon table.
wpn_clmns_to_lv=['offense_id', 'weapon']

# Due to the fact that sqlite has a limitation of not being able to drop␣
 ↪columns,
# I need to create a new table with only the columns I need, drop the old table␣
 ↪and rename the new one.
df=create_new_table('weapon_main', 'weapon_main_tmp', wpn_clmns_to_lv, cur,␣
 ↪drop_rename=True)
```

[32]:

```
q='SELECT * FROM weapon_main'
df=table_query(q, cur)
df.groupby('weapon').nunique()
```

[32]:

```
                       offense_id
weapon
Automatic firearm            2679
Non-automatic firearm      424464
Other weapon               107672
Unarmed                      2803
Unknown                     10263
```

Dropping unneeded tables

[33]:
```
# Dropping all the original incident, offense, offender, victim and weapon␣
 ↪tables

table_list_to_drop=['nibrs_victim','nibrs_offense','nibrs_incident','nibrs_weapon','nibrs_offe

for table in table_list_to_drop:
    string=table
    statement='DROP TABLE'+' '+string
    cur.execute(statement)
cur.execute("""SELECT name FROM sqlite_master WHERE type='table'""").fetchall()
```

[33]:
```
[('agencies',),
 ('agency_participation',),
 ('cde_agencies',),
 ('nibrs_age',),
 ('nibrs_bias_list',),
 ('nibrs_location_type',),
 ('nibrs_offense_type',),
 ('nibrs_victim_type',),
 ('nibrs_cleared_except',),
 ('nibrs_ethnicity',),
 ('nibrs_relationship',),
 ('nibrs_weapon_type',),
 ('ref_race',),
 ('nibrs_bias_motivation',),
 ('nibrs_victim_offender_rel',),
 ('incident_main',),
 ('offense_main',),
 ('offender_main',),
 ('victim_main',),
 ('weapon_main',)]
```

[34]:
```
# Dropping all obsolete reference tables
table_list_to_drop=['nibrs_age','nibrs_victim_type','nibrs_ethnicity','ref_race',␣
 ↪'nibrs_weapon_type']

for table in table_list_to_drop:
    string=table
    statement='DROP TABLE'+' '+string
    cur.execute(statement)
cur.execute("""SELECT name FROM sqlite_master WHERE type='table'""").fetchall()
```

[34]:
```
[('agencies',),
 ('agency_participation',),
 ('cde_agencies',),
 ('nibrs_bias_list',),
 ('nibrs_location_type',),
 ('nibrs_offense_type',),
 ('nibrs_cleared_except',),
 ('nibrs_relationship',),
 ('nibrs_bias_motivation',),
 ('nibrs_victim_offender_rel',),
 ('incident_main',),
 ('offense_main',),
 ('offender_main',),
 ('victim_main',),
 ('weapon_main',)]
```

Uncomment the following 2 cells, run them and comment out again if you want to re-run the code above.

[35]:

```
cur.close()
conn.commit()
conn.close()
```

[36]:

```
# !cp data/sqlite/db/production1_backup.db data/sqlite/db/production1.db

# !rm data/sqlite/db/production1_backup.db
```

At this point victim_main, offender_main and weapon_main tables are ready. I am creating an intermediate database to avoid the need to recreate the main one if I make a mistake.

3.1.2 Agencies

[37]:

```
# stmnt="DROP TABLE table_name"
# cur.execute(stmnt)
```

The cell below is to close a production1 db/cursor (commit too) and to use production1 db as a spring board moving forward. Uncomment the cell, run it to copy production1 to production2 plus production2 backup and comment it out again

[38]:

```
!cp data/sqlite/db/production1.db data/sqlite/db/production2.db
!cp data/sqlite/db/production2.db data/sqlite/db/production2_backup.db
```

[39]:

```
# Initiating a cursor
conn = sqlite3.connect('data/sqlite/db/production2.db')
cur = conn.cursor()
```

[40]:

```
cur.execute("""SELECT name FROM sqlite_master WHERE type='table'""").fetchall()
```

[40]:

```
[('agencies',),
 ('agency_participation',),
 ('cde_agencies',),
 ('nibrs_bias_list',),
 ('nibrs_location_type',),
 ('nibrs_offense_type',),
 ('nibrs_cleared_except',),
 ('nibrs_relationship',),
 ('nibrs_bias_motivation',),
 ('nibrs_victim_offender_rel',),
 ('incident_main',),
 ('offense_main',),
 ('offender_main',),
 ('victim_main',),
 ('weapon_main',)]
```

[41]:

```
# Checking if production1 copied correctly into production2
q='SELECT * FROM weapon_main'
df=table_query(q, cur)
df.groupby('weapon').nunique()
```

[41]:

```
                         offense_id
weapon
Automatic firearm              2679
Non-automatic firearm        424464
Other weapon                 107672
Unarmed                        2803
Unknown                       10263
```

agencies table

preparing agencies table before comparing it to cde_agencies table

[42]:

```
q='SELECT * from agencies'
df=table_query(q, cur)
df.columns
```

[42]:

```
Index(['yearly_agency_id', 'agency_id', 'data_year', 'ori', 'legacy_ori',
       'covered_by_legacy_ori', 'direct_contributor_flag', 'dormant_flag',
       'dormant_year', 'reporting_type', 'ucr_agency_name', 'ncic_agency_name',
       'pub_agency_name', 'pub_agency_unit', 'agency_status', 'state_id',
       'state_name', 'state_abbr', 'state_postal_abbr', 'division_code',
       'division_name', 'region_code', 'region_name', 'region_desc',
       'agency_type_name', 'population', 'submitting_agency_id', 'sai',
       'submitting_agency_name', 'suburban_area_flag', 'population_group_id',
       'population_group_code', 'population_group_desc',
       'parent_pop_group_code', 'parent_pop_group_desc', 'mip_flag',
       'pop_sort_order', 'summary_rape_def', 'pe_reported_flag',
       'male_officer', 'male_civilian', 'male_total', 'female_officer',
       'female_civilian', 'female_total', 'officer_rate', 'employee_rate',
       'nibrs_cert_date', 'nibrs_start_date', 'nibrs_leoka_start_date',
       'nibrs_ct_start_date', 'nibrs_multi_bias_start_date',
       'nibrs_off_eth_start_date', 'covered_flag', 'county_name', 'msa_name',
       'publishable_flag', 'participated', 'nibrs_participated'],
      dtype='object')
```

[43]:

```
df.head()
```

[43]:

```
   yearly_agency_id  agency_id  data_year        ori legacy_ori  \
0          18262016       1826       2016  CO0010000  CO0010000
1          18272016       1827       2016  CO0010100  CO0010100
2          18282016       1828       2016  CO0010200  CO0010200
3          18292016       1829       2016  CO0010300  CO0010300
4          18302016       1830       2016  CO0010400  CO0010400

  covered_by_legacy_ori direct_contributor_flag dormant_flag dormant_year  \
0                                              N            N
1                                              N            N
2                                              N            N
3                                              N            N
4                                              N            N

  reporting_type  ... nibrs_leoka_start_date nibrs_ct_start_date  \
0              I  ...              01-MAR-03           01-FEB-14
1              I  ...              01-MAR-03           01-FEB-14
2              I  ...              01-JAN-06           01-FEB-14
3              I  ...              01-MAR-03           01-FEB-14
4              I  ...              01-SEP-12           01-JUL-14

  nibrs_multi_bias_start_date nibrs_off_eth_start_date covered_flag  \
0                   01-JAN-16                01-APR-13            N
1                   01-JAN-16                01-APR-13            N
2                   01-JAN-16                01-APR-13            N
3                   01-JAN-16                01-APR-13            N
4                   01-FEB-16                01-APR-13            N

               county_name                                 msa_name  \
0                    ADAMS               Denver-Aurora-Lakewood, CO
1  DOUGLAS; ADAMS; ARAPAHOE               Denver-Aurora-Lakewood, CO
2              WELD; ADAMS  Denver-Aurora-Lakewood, CO; Greeley, CO
3                    ADAMS               Denver-Aurora-Lakewood, CO
4                    ADAMS               Denver-Aurora-Lakewood, CO

  publishable_flag participated  nibrs_participated
0                Y            Y                   Y
1                Y            Y                   Y
2                Y            Y                   Y
3                Y            Y                   Y
4                Y            Y                   Y
```

```
[5 rows x 59 columns]
```

```
[44]: # Dropping all unused columns
agncs_to_lv_agnctbl=['agency_id', 'data_year',
       'pub_agency_name',
        'county_name']

df=create_new_table('agencies', 'agencies_tmp', agncs_to_lv_agnctbl, cur,␣
 ↪drop_rename=True)
```

```
[45]: q='SELECT * from agencies'
df=table_query(q, cur)
df.head()
```

```
[45]:    agency_id  data_year pub_agency_name                 county_name
0        1826       2016           Adams                       ADAMS
1        1827       2016          Aurora  DOUGLAS; ADAMS; ARAPAHOE
2        1828       2016        Brighton                 WELD; ADAMS
3        1829       2016   Commerce City                       ADAMS
4        1830       2016        Thornton                       ADAMS
```

```
[46]: df['agency_id'].nunique()
```

```
[46]: 236
```

cde_agencies table

Preparing cde_agencies table befor comparing it to agencies table

```
[47]: q='SELECT * from cde_agencies'
df=table_query(q, cur)
df.head()
```

```
[47]:    agency_id        ori legacy_ori                       agency_name  \
0        1904  CO0180000  CO0180000   Douglas County Sheriff's Office
1        1995  CO0370100  CO0370100           Limon Police Department
2        1954  CO0280000  CO0280000  Huerfano County Sheriff's Office
3        1937  CO0230500  CO0230500            Silt Police Department
4        1870  CO0070800  CO0070800       Nederland Police Department

  short_name  agency_type_id agency_type_name tribe_id campus_id city_id  …  \
0    Douglas               2           County                              …
1      Limon               1             City                       1135   …
2   Huerfano               2           County                              …
3       Silt               1             City                       1186   …
4  Nederland               1             City                       1156   …

  past_10_years_reported  covered_by_id covered_by_ori  covered_by_name  \
0                     10
1                     10
2                      7
3                     10
4                      5

  staffing_year total_officers total_civilians  icpsr_zip icpsr_lat icpsr_lng
0          2016            309             161      80109   39.3264  -104.926
1          2016              5               1      80828   38.9937  -103.508
2          2016             10              13      81089   37.6878   -104.96
3          2016              6               0      81652   39.5994   -107.91
4          2016              5               1      80466   40.0948  -105.398

[5 rows x 44 columns]
```

[48]:
```
df.columns
```

[48]:
```
Index(['agency_id', 'ori', 'legacy_ori', 'agency_name', 'short_name',
       'agency_type_id', 'agency_type_name', 'tribe_id', 'campus_id',
       'city_id', 'city_name', 'state_id', 'state_abbr', 'primary_county_id',
       'primary_county', 'primary_county_fips', 'agency_status',
       'submitting_agency_id', 'submitting_sai', 'submitting_name',
       'submitting_state_abbr', 'start_year', 'dormant_year', 'current_year',
       'revised_rape_start', 'current_nibrs_start_year', 'population',
       'population_group_code', 'population_group_desc',
       'population_source_flag', 'suburban_area_flag', 'core_city_flag',
       'months_reported', 'nibrs_months_reported', 'past_10_years_reported',
       'covered_by_id', 'covered_by_ori', 'covered_by_name', 'staffing_year',
       'total_officers', 'total_civilians', 'icpsr_zip', 'icpsr_lat',
       'icpsr_lng'],
      dtype='object')
```

[49]:
```
# Dropping all the columns that seem to be irrelevant. Long and lat coordinates␣
↪are useless due to the fact that they are
# either of a center of a zipcode or a center of a county. Either way is't␣
↪useless

agncs_to_lv_cdeagnctbl=['agency_id', 'agency_name', 'short_name',
       'primary_county_id',
       'primary_county',
       'current_year',
       'icpsr_zip']

df=create_new_table('cde_agencies', 'cde_agencies_tmp', agncs_to_lv_cdeagnctbl,␣
↪cur, drop_rename=True)
```

[50]:

```
q='SELECT * from cde_agencies'
df=table_query(q, cur)
df.head()
```

[50]:

```
   agency_id                      agency_name short_name  primary_county_id  \
0       1904   Douglas County Sheriff's Office    Douglas                273
1       1995            Limon Police Department      Limon                292
2       1954  Huerfano County Sheriff's Office   Huerfano                283
3       1937             Silt Police Department       Silt                278
4       1870        Nederland Police Department  Nederland                261

  primary_county  current_year icpsr_zip
0        Douglas          2016     80109
1        Lincoln          2016     80828
2       Huerfano          2016     81089
3       Garfield          2016     81652
4        Boulder          2016     80466
```

Comparing cde_agencies and agencies tables to use one of them moving forward

[51]:

```
df['agency_id'].nunique()
```

[51]:

```
304
```

[52]:

```
q="SELECT distinct(agency_id) FROM agencies where agency_ID not in (SELECT␣
 ↪agency_id FROM cde_agencies)"
df=table_query(q, cur)
df
```

[52]:

```
   agency_id
0      29074
```

[53]:

```
stmnt="SELECT * FROM agencies where agency_ID=29074"
df = pd.DataFrame(cur.execute(stmnt))
df
```

[53]:

```
       0     1                                                   2          3
0  29074  2018  Division of Gaming Criminal Enforcement and In…  JEFFERSON
1  29074  2019  Division of Gaming Criminal Enforcement and In…  JEFFERSON
```

[54]:

```
stmnt="SELECT distinct(agency_id) FROM incident_main where agency_id not in␣
 ↪(SELECT agency_id FROM cde_agencies)"
df = pd.DataFrame(cur.execute(stmnt))
df
```

[54]:

```
Empty DataFrame
Columns: []
Index: []
```

```
[55]: clmns_to_lv_cdeagnctbl=['agency_id',
                             'primary_county',
                             'icpsr_zip']

      df=create_new_table('cde_agencies', 'cde_agencies_tmp', clmns_to_lv_cdeagnctbl,␣
       ↪cur, drop_rename=True)
```

Conclusion

> There are more counties (and their names are spelled out rather than merged together) in **cde_agencies**. Also there are zip codes in **cde_agencies**. There are 223 zip codes out of 511 active zip codes in Colorado. * There are 14 agencies that have records in incident_main table but are missing from agencies table while they are present in **cde_agencies**. * There is one agency (agency_id=29074), it is a Division of Gaming Criminal Enforcement in Jefferson county, that is in **agencies** table but is not in **cde_agencies**. However, this agency has no incident records.

The final conclusion that only **cde_agencies** table will be used moving forward.

3.1.3 Other tables

There are cleaned-up tables: * cde_agencies * incident_main * offense_main * victim_main * offender_main * weapon_main

There are tables that need to be cleaned and joined with the main tables: * nibrs_bias_list * nibrs_location_type * nibrs_offense_type * nibrs_cleared_except * nibrs_relationship * nibrs_bias_motivation * nibrs_victim_offender_rel

There are several tables that need to be deleted: * agencies * agency_participation * nibrs_criminal_act * nibrs_criminal_act_type * nibrs_victim_offense > Agencies and agency_participation are being dropped as explained above.

```
[56]: # Deleting the tables above

      table_list_to_drop=['agencies','agency_participation']

      for table in table_list_to_drop:
          string=table
          statement='DROP TABLE'+' '+string
          cur.execute(statement)
      cur.execute("""SELECT name FROM sqlite_master WHERE type='table'""").fetchall()
```

```
[56]: [('nibrs_bias_list',),
       ('nibrs_location_type',),
       ('nibrs_offense_type',),
       ('nibrs_cleared_except',),
       ('nibrs_relationship',),
       ('nibrs_bias_motivation',),
       ('nibrs_victim_offender_rel',),
       ('incident_main',),
       ('offense_main',),
       ('offender_main',),
       ('victim_main',),
       ('weapon_main',),
       ('cde_agencies',)]
```

Bias table

Adding bias type info to the main bias table

[57]:

```
q="SELECT * FROM nibrs_bias_list"
df = table_query(q, cur)
df
```

[57]:

```
    bias_id bias_code                                          bias_name
0        23        16       Anti-Native Hawaiian or Other Pacific Islander
1        24        51                              Anti-Physical Disability
2        25        52                                Anti-Mental Disability
3        26        61                                             Anti-Male
4        27        62                                           Anti-Female
5        28        71                                      Anti-Transgender
6        29        72                           Anti-Gender Non-Conforming
7         1        11                                            Anti-White
8         2        12                      Anti-Black or African American
9         3        13                 Anti-American Indian or Alaska Native
10        4        14                                            Anti-Asian
11        5        15                               Anti-Multi-Racial Group
12        6        21                                           Anti-Jewish
13        7        22                                         Anti-Catholic
14        8        23                                       Anti-Protestant
15        9        24                                 Anti-Islamic (Muslem)
16       10        25                                   Anti-Other Religion
17       11        26                           Anti-Multi-Religious Group
18       12        27                              Anti-Atheist/Agnosticism
19       13        31                                             Anti-Arab
20       14        32                               Anti-Hispanic or Latino
21       15        33                           Anti-Not Hispanic or Latino
22       16        41                            Anti-Male Homosexual (Gay)
23       17        42                      Anti-Female Homosexual (Lesbian)
24       18        43   Anti-Lesbian, Gay, Bisexual, or Transgender, M…
25       19        44                                  Anti-Heterosexual
26       20        45                                         Anti-Bisexual
27       21        88                                                  None
28       22        99                                               Unknown
29       30        28                                           Anti-Mormon
30       31        29                                Anti-Jehovah's Witness
31       32        81                                 Anti-Eastern Orthodox
```

```
32        33         82                                Anti-Other Christian
33        34         83                                       Anti-Buddhist
34        35         84                                          Anti-Hindu
35        36         85                                           Anti-Sikh
```

[58]:

```
# Intermediatly (to be dropped later) adding 'bias_name' column to bias_main␣
 ↪table

bias_clmns_to_lv=['bias_id', 'offense_id']

# Due to the fact that sqlite has a limitation of not being able to drop␣
 ↪columns,
# I need to create a new table with only the columns I need.

create_new_table('nibrs_bias_motivation', 'bias_main', bias_clmns_to_lv, cur)

add_update_clmn('bias_main','nibrs_bias_list', 'bias_name', 'bias_name',␣
 ↪'bias_id', cur)
```

[58]:

```
         bias_id  offense_id bias_name
0             21    53563151      None
1             21    53563402      None
2             21    53558278      None
3             21    53558279      None
4             21    53563403      None
…          …          …        …
3201153       21   132477865      None
3201154       21   132483473      None
3201155       21   132486411      None
3201156       21   132486743      None
3201157       21   132485724      None

[3201158 rows x 3 columns]
```

[59]:

```
# Making sure the columns are there
q='SELECT * FROM bias_main'
df=table_query(q, cur)
df.bias_name.unique()
```

[59]:

```
array(['None', 'Anti-Black or African American', 'Anti-White',
       'Anti-Physical Disability', 'Anti-Hispanic or Latino',
       'Anti-Not Hispanic or Latino', 'Anti-Female Homosexual (Lesbian)',
       'Anti-Asian',
       'Anti-Lesbian, Gay, Bisexual, or Transgender, Mixed Group (LGBT)',
       'Anti-Jewish', 'Anti-Male Homosexual (Gay)',
       'Anti-American Indian or Alaska Native', 'Anti-Catholic',
       'Anti-Multi-Racial Group', 'Anti-Mental Disability',
       'Anti-Islamic (Muslem)', 'Anti-Other Religion',
       'Anti-Multi-Religious Group', 'Unknown', 'Anti-Protestant',
       'Anti-Bisexual', 'Anti-Heterosexual', 'Anti-Atheist/Agnosticism',
       'Anti-Transgender', 'Anti-Other Christian', 'Anti-Arab',
       "Anti-Jehovah's Witness", 'Anti-Female',
       'Anti-Gender Non-Conforming', 'Anti-Buddhist'], dtype=object)
```

[60]:

```
bias_to_lv_biasmot=['offense_id',
       'bias_name']

df=create_new_table('bias_main', 'bias_main_tmp', bias_to_lv_biasmot, cur,␣
 ↪drop_rename=True)
```

[61]:

```
q='SELECT * FROM bias_main'
df=table_query(q, cur)
df.groupby('bias_name').nunique()
```

[61]:

```
                                                     offense_id
bias_name
Anti-American Indian or Alaska Native                        30
Anti-Arab                                                     8
Anti-Asian                                                   25
Anti-Atheist/Agnosticism                                      2
Anti-Bisexual                                                10
Anti-Black or African American                              426
Anti-Buddhist                                                 1
Anti-Catholic                                                11
Anti-Female                                                   1
Anti-Female Homosexual (Lesbian)                             47
Anti-Gender Non-Conforming                                    1
Anti-Heterosexual                                             1
Anti-Hispanic or Latino                                     214
Anti-Islamic (Muslem)                                        50
Anti-Jehovah's Witness                                        3
Anti-Jewish                                                 106
Anti-Lesbian, Gay, Bisexual, or Transgender, Mi…          128
Anti-Male Homosexual (Gay)                                  162
Anti-Mental Disability                                       11
Anti-Multi-Racial Group                                      48
Anti-Multi-Religious Group                                   19
Anti-Not Hispanic or Latino                                  63
Anti-Other Christian                                          4
Anti-Other Religion                                          27
Anti-Physical Disability                                     16
Anti-Protestant                                              17
Anti-Transgender                                             12
Anti-White                                                  169
```

```
None                                                     3199416
Unknown                                                      130
```

Location in the offense table

Leaving all location types in. However, I might reconsider later to change to Home/Residence, Other and Unknown only

[62]:
```
# Adding a new column to offense table with location_names

add_update_clmn('offense_main','nibrs_location_type', 'location_name',␣
↪'location_name', 'location_id', cur)

q='SELECT * FROM offense_main'
df=table_query(q, cur)
df.location_name.unique()
```

[62]:
```
array(['Residence/Home', 'School/College', 'Other/Unknown',
       'Service/Gas Station', 'Commercial/Office Building',
       'Department/Discount Store', 'Jail/Prison', 'Field/Woods',
       'Highway/Road/Ally', 'Government/Public Building',
       'Convenience Store', 'Parking Lot/Garage', 'Hotel/Motel/Etc.',
       'Bar/Nightclub', 'Liquor Store', 'Air/Bus/Train Terminal',
       'Rental Stor. Facil.', 'Drug Store/Dr. s Office/Hospital',
       'Construction Site', 'Specialty Store', 'Grocery/Supermarket',
       'Bank/Savings and Loan', 'Restaurant', 'Church Synagogue/Temple',
       'Lake/Waterway', 'School-Elementary/Secondary', 'Industrial Site',
       'Park/Playground', 'Auto Dealership New/Used',
       'School-College/University', 'Shopping Mall', 'Camp/Campground',
       'Dock/Wharf/Freight/Modal Terminal', 'Farm Facility',
       'Amusement Park', 'Gambling Facility/Casino/Race Track',
       'Abandoned/Condemned Structure',
       'Arena/Stadium/Fairgrounds/Coliseum', 'Shelter-Mission/Homeless',
       'ATM Separate from Bank', 'Daycare Facility', 'Rest Area',
       'Military Installation', 'Tribal Lands', 'Community Center',
       'Cyberspace'], dtype=object)
```

[63]:
```
df.groupby('location_name').nunique()
```

[63]:
```
                                     offense_id  incident_id  offense_type_id  \
location_name
ATM Separate from Bank                     1156         1018               29
Abandoned/Condemned Structure               734          623               30
Air/Bus/Train Terminal                    12132        11537               40
Amusement Park                             1062          989               34
Arena/Stadium/Fairgrounds/Coliseum         1995         1846               34
Auto Dealership New/Used                   5926         5158               36
Bank/Savings and Loan                     31810        25871               37
```

```
Bar/Nightclub                             32853        30359               45
Camp/Campground                            1555         1353               35
Church Synagogue/Temple                    9121         8185               40
Commercial/Office Building                56070        50351               46
Community Center                           4230         3880               38
Construction Site                         20817        18551               36
Convenience Store                         50154        45250               46
Cyberspace                                 3395         2922               18
Daycare Facility                           1075         1010               33
Department/Discount Store                198684       180624               44
Dock/Wharf/Freight/Modal Terminal           582          543               27
Drug Store/Dr. s Office/Hospital          30523        27818               45
Farm Facility                              1487         1303               32
Field/Woods                               19348        17574               43
Gambling Facility/Casino/Race Track        3259         2948               37
Government/Public Building                26425        24250               44
Grocery/Supermarket                       71688        66204               43
Highway/Road/Ally                        484729       419285               49
Hotel/Motel/Etc.                          51263        43426               47
Industrial Site                            3672         3076               33
Jail/Prison                               18809        17807               39
Lake/Waterway                              1169         1035               32
Liquor Store                              13177        11780               40
Military Installation                       122          110               22
Other/Unknown                            172321       158785               50
Park/Playground                           25124        22156               46
Parking Lot/Garage                       384128       342816               50
Rental Stor. Facil.                       17790        15143               39
Residence/Home                          1156469      1029236               50
Rest Area                                   361          320               29
Restaurant                                51034        46226               44
School-College/University                 31454        27295               43
School-Elementary/Secondary               52122        46659               42
School/College                            35013        32177               40
Service/Gas Station                       20883        18670               41
Shelter-Mission/Homeless                   1086         1023               33
Shopping Mall                              7332         6436               39
Specialty Store                           86896        78668               46
Tribal Lands                                108          104               21

                                     location_id
location_name
ATM Separate from Bank                         1
Abandoned/Condemned Structure                  1
Air/Bus/Train Terminal                         1
Amusement Park                                 1
Arena/Stadium/Fairgrounds/Coliseum             1
```

```
Auto Dealership New/Used                       1
Bank/Savings and Loan                          1
Bar/Nightclub                                  1
Camp/Campground                                1
Church Synagogue/Temple                        1
Commercial/Office Building                     1
Community Center                               1
Construction Site                              1
Convenience Store                              1
Cyberspace                                     1
Daycare Facility                               1
Department/Discount Store                      1
Dock/Wharf/Freight/Modal Terminal              1
Drug Store/Dr. s Office/Hospital               1
Farm Facility                                  1
Field/Woods                                    1
Gambling Facility/Casino/Race Track            1
Government/Public Building                     1
Grocery/Supermarket                            1
Highway/Road/Ally                              1
Hotel/Motel/Etc.                               1
Industrial Site                                1
Jail/Prison                                    1
Lake/Waterway                                  1
Liquor Store                                   1
Military Installation                          1
Other/Unknown                                  1
Park/Playground                                1
Parking Lot/Garage                             1
Rental Stor. Facil.                            1
Residence/Home                                 1
Rest Area                                      1
Restaurant                                     1
School-College/University                      1
School-Elementary/Secondary                    1
School/College                                 1
Service/Gas Station                            1
Shelter-Mission/Homeless                       1
Shopping Mall                                  1
Specialty Store                                1
Tribal Lands                                   1
```

[64]:

```
df.nunique()
```

[64]:

```
offense_id         3201143
incident_id        2819189
offense_type_id         51
```

```
location_id            46
location_name          46
dtype: int64
```

Offense type in the offense table

Adding offense type info to the main offense table

```
[65]: q='SELECT * from nibrs_offense_type'
df=table_query(q, cur)
df
```

```
[65]:     offense_type_id offense_code                              offense_name  \
0                58          23*                             Not Specified
1                 1          09C                      Justifiable Homicide
2                 2          26A   False Pretenses/Swindle/Confidence Game
3                 3          36B                            Statutory Rape
4                 4          11C             Sexual Assault With An Object
..              …          …                                       …
59               60          64B  Human Trafficking, Involuntary Servitude
60               61          40C                   Purchasing Prostitution
61               63          26F                            Identity Theft
62               64          26G                 Hacking/Computer Invasion
63               62          720                            Animal Cruelty

   crime_against ct_flag hc_flag hc_code   offense_category_name
0       Property       N       Y      06  Larceny/Theft Offenses
1    Not a Crime       N       N               Homicide Offenses
2       Property       Y       Y                  Fraud Offenses
3         Person       N       Y                    Sex Offenses
4         Person       N       Y      02            Sex Offenses
..           …     …     …       …                     …
59        Person       N       Y               Human Trafficking
60       Society       N       Y           Prostitution Offenses
61      Property       N       Y                  Fraud Offenses
62      Property       N       Y                  Fraud Offenses
63       Society       N       N                  Animal Cruelty

[64 rows x 8 columns]
```

```
[66]: # Adding a new column to offense table with offense_type name

add_update_clmn('offense_main','nibrs_offense_type', 'offense_name',␣
 ↪'offense_name', 'offense_type_id', cur)

add_update_clmn('offense_main','nibrs_offense_type', 'crime_against',␣
 ↪'crime_against', 'offense_type_id', cur)
```

```
add_update_clmn('offense_main','nibrs_offense_type', 'offense_category_name',␣
↪'offense_category_name',
                'offense_type_id', cur)
```

[66]:

```
         offense_id  incident_id  offense_type_id  location_id  \
0          53563151     51264520               27           20
1          53563402     51264521               14           20
2          53558278     51264523               16           22
3          53558279     51264523               35           22
4          53563403     51264524               46           25
…             …          …              …          …
3201138   141844716    116813642                5           18
3201139   141852632    116813645               35            8
3201140   141848922    116813645               16            8
3201141   141844745    116813666               16           38
3201142   141848949    116813669               49           20

                     location_name                              offense_name  \
0                   Residence/Home                        Aggravated Assault
1                   Residence/Home                  Theft From Motor Vehicle
2                   School/College                  Drug/Narcotic Violations
3                   School/College                 Drug Equipment Violations
4                    Other/Unknown                             Impersonation
…                            …                                       …
3201138         Parking Lot/Garage  Destruction/Damage/Vandalism of Property
3201139  Department/Discount Store                 Drug Equipment Violations
3201140  Department/Discount Store                  Drug/Narcotic Violations
3201141            Park/Playground                  Drug/Narcotic Violations
3201142             Residence/Home              Burglary/Breaking & Entering

        crime_against                     offense_category_name
0              Person                          Assault Offenses
1            Property                    Larceny/Theft Offenses
2             Society                    Drug/Narcotic Offenses
3             Society                    Drug/Narcotic Offenses
4            Property                            Fraud Offenses
…               …                                       …
3201138      Property  Destruction/Damage/Vandalism of Property
3201139       Society                    Drug/Narcotic Offenses
3201140       Society                    Drug/Narcotic Offenses
3201141       Society                    Drug/Narcotic Offenses
3201142      Property              Burglary/Breaking & Entering

[3201143 rows x 8 columns]
```

[67]:

```
# Dropping all unused columns
offns_to_lv_offnstbl=['offense_id',␣
 ↪'incident_id','location_name','offense_name','crime_against','offense_category_name']

df=create_new_table('offense_main', 'offense_main_tmp', offns_to_lv_offnstbl,␣
 ↪cur, drop_rename=True)
```

[68]:

```
q='SELECT * from offense_main'
df=table_query(q, cur)
df.head()
```

[68]:

```
   offense_id  incident_id   location_name               offense_name  \
0    53563151     51264520  Residence/Home         Aggravated Assault
1    53563402     51264521  Residence/Home   Theft From Motor Vehicle
2    53558278     51264523  School/College   Drug/Narcotic Violations
3    53558279     51264523  School/College  Drug Equipment Violations
4    53563403     51264524   Other/Unknown              Impersonation

  crime_against   offense_category_name
0        Person        Assault Offenses
1      Property  Larceny/Theft Offenses
2       Society  Drug/Narcotic Offenses
3       Society  Drug/Narcotic Offenses
4      Property          Fraud Offenses
```

Victim-offender relationship

Adding victim-offender relationship info to the main victim table

[69]:

```
cur.execute("""SELECT name FROM sqlite_master WHERE type='table'""").fetchall()
```

[69]:

```
[('nibrs_bias_list',),
 ('nibrs_location_type',),
 ('nibrs_offense_type',),
 ('nibrs_cleared_except',),
 ('nibrs_relationship',),
 ('nibrs_bias_motivation',),
 ('nibrs_victim_offender_rel',),
 ('incident_main',),
 ('offender_main',),
 ('victim_main',),
 ('weapon_main',),
 ('cde_agencies',),
 ('bias_main',),
 ('offense_main',)]
```

[70]:

```
q='SELECT * from nibrs_relationship'
df=table_query(q, cur)
```

```
df.head()
```

```
[70]:    relationship_id relationship_code  \
      0                1                AQ
      1                2                BE
      2                3                BG
      3                4                CF
      4                5                CH

                                  relationship_name
      0                     Victim Was Acquaintance
      1                       Victim Was Babysittee
      2             Victim Was Boyfriend/Girlfriend
      3  Victim Was Child of Boyfriend or Girlfriend
      4                            Victim Was Child
```

```
[71]: q='SELECT * from nibrs_victim_offender_rel'
      df=table_query(q, cur)
      df.head()
```

```
[71]:    victim_id  offender_id  relationship_id  nibrs_victim_offender_id
      0   55514644     57702592               16                  16117589
      1   55514649     57702597               20                  15965036
      2   55514652     57702601               21                  15965035
      3   55514653     57702602                3                  15965034
      4   55514655     57702604                5                  15965033
```

```
[72]: add_update_clmn('nibrs_victim_offender_rel','nibrs_relationship',␣
       ↪'relationship_name', 'relationship_name',
                     'relationship_id', cur)
```

```
[72]:          victim_id  offender_id  relationship_id  nibrs_victim_offender_id  \
      0        55514644     57702592               16                  16117589
      1        55514649     57702597               20                  15965036
      2        55514652     57702601               21                  15965035
      3        55514653     57702602                3                  15965034
      4        55514655     57702604                5                  15965033
      …             …            …                …                         …
      794152  128903173    133669903               24                  40271007
      794153  128898322    133669913               24                  40261336
      794154  128897289    133685015                3                  40271074
      794155  128897328    133680303               21                  40271089
      794156  128898519    133685096               16                  40271100

                       relationship_name
      0           Victim was Otherwise Known
      1                Victim Was Stepchild
```

```
2                      Victim Was Spouse
3        Victim Was Boyfriend/Girlfriend
4                       Victim Was Child
…                                      …
794152                Victim Was Stranger
794153                Victim Was Stranger
794154   Victim Was Boyfriend/Girlfriend
794155                  Victim Was Spouse
794156        Victim was Otherwise Known

[794157 rows x 5 columns]
```

```
[73]: # Dropping all unused columns
      clmns_to_lv_rlshnshptbl=['victim_id', 'offender_id','relationship_name']

      df=create_new_table('nibrs_victim_offender_rel',␣
       ↪'nibrs_victim_offender_rel_tmp',
                          clmns_to_lv_rlshnshptbl, cur, drop_rename=True)
```

```
[74]: q='SELECT * from nibrs_victim_offender_rel'
      df=table_query(q, cur)
      df.head()
```

```
[74]:    victim_id  offender_id                relationship_name
      0   55514644     57702592       Victim was Otherwise Known
      1   55514649     57702597             Victim Was Stepchild
      2   55514652     57702601                Victim Was Spouse
      3   55514653     57702602  Victim Was Boyfriend/Girlfriend
      4   55514655     57702604                 Victim Was Child
```

```
[75]: stmnt='ALTER TABLE nibrs_victim_offender_rel RENAME to victim_offender_rel'
      cur.execute(stmnt)
```

```
[75]: <sqlite3.Cursor at 0x1cb6f69f880>
```

```
[76]: cur.execute("""SELECT name FROM sqlite_master WHERE type='table'""").fetchall()
```

```
[76]: [('nibrs_bias_list',),
       ('nibrs_location_type',),
       ('nibrs_offense_type',),
       ('nibrs_cleared_except',),
       ('nibrs_relationship',),
       ('nibrs_bias_motivation',),
       ('incident_main',),
       ('offender_main',),
       ('victim_main',),
       ('weapon_main',),
       ('cde_agencies',),
       ('bias_main',),
       ('offense_main',),
       ('victim_offender_rel',)]
```

Dropping all reference tables

[77]:
```
table_list_to_drop=['nibrs_bias_list',
                    'nibrs_location_type',
                    'nibrs_offense_type',
                    'nibrs_cleared_except',
                    'nibrs_relationship',
                    'nibrs_bias_motivation']

for table in table_list_to_drop:
    string=table
    statement='DROP TABLE'+' '+string
    cur.execute(statement)
cur.execute("""SELECT name FROM sqlite_master WHERE type='table'""").fetchall()
```

[77]:
```
[('incident_main',),
 ('offender_main',),
 ('victim_main',),
 ('weapon_main',),
 ('cde_agencies',),
 ('bias_main',),
 ('offense_main',),
 ('victim_offender_rel',)]
```

3.1.4 Combining all tables into one based on offense table

Incident table

Adding agencies info into the main incident table and dropping the cde_agencies table. Replacing ’ ’ in the incident table hour column to ‘0’.

[78]:
```
q='SELECT * from incident_main'
df=table_query(q, cur)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2819463 entries, 0 to 2819462
Data columns (total 4 columns):
 #   Column         Dtype
---  ------         -----
 0   agency_id      int64
 1   incident_id    int64
 2   incident_date  object
 3   incident_hour  object
```

```
dtypes: int64(2), object(2)
memory usage: 86.0+ MB
```

```
[79]: q='SELECT * from cde_agencies'
      df=table_query(q, cur)
      df
```

```
[79]:       agency_id primary_county icpsr_zip
      0          1904        Douglas     80109
      1          1995        Lincoln     80828
      2          1954       Huerfano     81089
      3          1937       Garfield     81652
      4          1870        Boulder     80466
      …          …             …        …
      2099       1828          Adams     80601
      2100       1904        Douglas     80109
      2101       1842       Arapahoe     80110
      2102       1963      Jefferson     80033
      2103       2039           Park     80420

      [2104 rows x 3 columns]
```

```
[80]: remove_dups('cde_agencies', 'cde_agencies_nodups', conn, cur, drop_rename=True)
```

```
[80]:      index  agency_id primary_county icpsr_zip
      0        0       1904        Douglas     80109
      1        1       1995        Lincoln     80828
      2        2       1954       Huerfano     81089
      3        3       1937       Garfield     81652
      4        4       1870        Boulder     80466
      ..     …          …             …        …
      299    778      23212           Weld     80642
      300    843      23131       Arapahoe
      301   1003      25267         Moffat
      302   1009      23240     San Miguel     81435
      303   1311      25314          Eagle

      [304 rows x 4 columns]
```

```
[81]: add_update_clmn('incident_main','cde_agencies', 'primary_county',␣
       ↪'primary_county', 'agency_id', cur)

      add_update_clmn('incident_main','cde_agencies', 'icpsr_zip', 'icpsr_zip',␣
       ↪'agency_id', cur)
```

```
[81]:          agency_id  incident_id        incident_date incident_hour  \
      0             1971     51264520  2009-01-05 00:00:00            22
      1             1971     51264521  2009-01-13 00:00:00
```

```
2             1971  51264523  2009-01-17 00:00:00             19
3             1971  51264524  2009-01-20 00:00:00
4             1971  51264525  2009-01-21 00:00:00
…            …          …                  …           …
2819458       2023 120337425            17-Dec-19              9
2819459       2023 119323671            21-Dec-19             14
2819460       2023 119323654            19-Dec-19             22
2819461       2023 120333220            13-Oct-19             13
2819462       2023 120337420            24-Nov-19             13

        primary_county icpsr_zip
0           Kit Carson     80807
1           Kit Carson     80807
2           Kit Carson     80807
3           Kit Carson     80807
4           Kit Carson     80807
…                …       …
2819458         Morgan     80701
2819459         Morgan     80701
2819460         Morgan     80701
2819461         Morgan     80701
2819462         Morgan     80701

[2819463 rows x 6 columns]
```

[82]:

```
q='SELECT * from incident_main'
df=table_query(q, cur)
df
```

[82]:

```
         agency_id  incident_id        incident_date incident_hour  \
0             1971  51264520  2009-01-05 00:00:00             22
1             1971  51264521  2009-01-13 00:00:00
2             1971  51264523  2009-01-17 00:00:00             19
3             1971  51264524  2009-01-20 00:00:00
4             1971  51264525  2009-01-21 00:00:00
…            …          …                  …           …
2819458       2023 120337425            17-Dec-19              9
2819459       2023 119323671            21-Dec-19             14
2819460       2023 119323654            19-Dec-19             22
2819461       2023 120333220            13-Oct-19             13
2819462       2023 120337420            24-Nov-19             13

        primary_county icpsr_zip
0           Kit Carson     80807
1           Kit Carson     80807
2           Kit Carson     80807
3           Kit Carson     80807
```

```
4            Kit Carson     80807
…                 …        …
2819458          Morgan     80701
2819459          Morgan     80701
2819460          Morgan     80701
2819461          Morgan     80701
2819462          Morgan     80701

[2819463 rows x 6 columns]
```

[83]:

```
df.incident_hour.isna().sum()
```

[83]:

```
0
```

[84]:

```
update_value('incident_main', 'incident_hour', "''", '25', cur)
```

[84]:

```
         agency_id  incident_id        incident_date  incident_hour  \
0             1971     51264520  2009-01-05 00:00:00             22
1             1971     51264521  2009-01-13 00:00:00             25
2             1971     51264523  2009-01-17 00:00:00             19
3             1971     51264524  2009-01-20 00:00:00             25
4             1971     51264525  2009-01-21 00:00:00             25
…              …          …                    …            …
2819458       2023    120337425            17-Dec-19              9
2819459       2023    119323671            21-Dec-19             14
2819460       2023    119323654            19-Dec-19             22
2819461       2023    120333220            13-Oct-19             13
2819462       2023    120337420            24-Nov-19             13

        primary_county icpsr_zip
0            Kit Carson     80807
1            Kit Carson     80807
2            Kit Carson     80807
3            Kit Carson     80807
4            Kit Carson     80807
…                 …        …
2819458          Morgan     80701
2819459          Morgan     80701
2819460          Morgan     80701
2819461          Morgan     80701
2819462          Morgan     80701

[2819463 rows x 6 columns]
```

[85]:

```
stmnt="DROP TABLE cde_agencies"
cur.execute(stmnt)
```

```
[85]: <sqlite3.Cursor at 0x1cb6f69f880>
```

Creating dataframes and saving them to pickle files to finalize working with sqlite tables

```
[86]: cur.execute("""SELECT name FROM sqlite_master WHERE type='table'""").fetchall()
```

```
[86]: [('incident_main',),
       ('offender_main',),
       ('victim_main',),
       ('weapon_main',),
       ('bias_main',),
       ('offense_main',),
       ('victim_offender_rel',)]
```

```
[87]: q='SELECT * from incident_main'
      df_incident=table_query(q, cur)
      with open('data/pickled_dataframes/incident.pickle', 'wb') as f:
          pickle.dump(df_incident, f)
```

```
[88]: with open('data/pickled_dataframes/incident.pickle', 'rb') as f:
          df_incident=pickle.load(f)
      df_incident.head()
```

```
[88]:    agency_id  incident_id        incident_date  incident_hour primary_county  \
      0       1971     51264520  2009-01-05 00:00:00             22     Kit Carson
      1       1971     51264521  2009-01-13 00:00:00             25     Kit Carson
      2       1971     51264523  2009-01-17 00:00:00             19     Kit Carson
      3       1971     51264524  2009-01-20 00:00:00             25     Kit Carson
      4       1971     51264525  2009-01-21 00:00:00             25     Kit Carson

        icpsr_zip
      0     80807
      1     80807
      2     80807
      3     80807
      4     80807
```

```
[89]: len(df_incident)
```

```
[89]: 2819463
```

```
[90]: q='SELECT * from offense_main'
      df_offense=table_query(q, cur)
      with open('data/pickled_dataframes/offense.pickle', 'wb') as f:
          pickle.dump(df_offense, f)
```

[91]:

```
with open('data/pickled_dataframes/offense.pickle', 'rb') as f:
    df_offense=pickle.load(f)
df_offense.head()
```

[91]:

```
   offense_id  incident_id   location_name              offense_name  \
0    53563151     51264520  Residence/Home        Aggravated Assault
1    53563402     51264521  Residence/Home    Theft From Motor Vehicle
2    53558278     51264523  School/College   Drug/Narcotic Violations
3    53558279     51264523  School/College  Drug Equipment Violations
4    53563403     51264524   Other/Unknown             Impersonation

  crime_against   offense_category_name
0        Person        Assault Offenses
1      Property  Larceny/Theft Offenses
2       Society  Drug/Narcotic Offenses
3       Society  Drug/Narcotic Offenses
4      Property          Fraud Offenses
```

[92]:

```
len(df_offense)
```

[92]:

```
3201143
```

[93]:

```
q='SELECT * from offender_main'
df_offender=table_query(q, cur)
with open('data/pickled_dataframes/offender.pickle', 'wb') as f:
    pickle.dump(df_offender, f)
```

[94]:

```
with open('data/pickled_dataframes/offender.pickle', 'rb') as f:
    df_offender=pickle.load(f)
df_offender.head()
```

[94]:

```
   offender_id  incident_id age_num sex_code   race     age_group ethnicity
0     57702592     51264520      25     Male  White  Age in Years      None
1     57702593     51264521                    None          None      None
2     57702595     51264523      20     Male  White  Age in Years      None
3     57702596     51264524                    None          None      None
4     57702597     51264525      55     Male  White  Age in Years      None
```

[95]:

```
len(df_offender)
```

[95]:

```
3197991
```

[96]:

```
q='SELECT * from victim_main'
df_victim=table_query(q, cur)
with open('data/pickled_dataframes/victim.pickle', 'wb') as f:
    pickle.dump(df_victim, f)
```

```
[97]: with open('data/pickled_dataframes/victim.pickle', 'rb') as f:
          df_victim=pickle.load(f)
      df_victim.head()
```

```
[97]:    victim_id  incident_id age_num sex_code resident_status_code   race  \
      0   55514644     51264520      23     Male             Resident  White
      1   55514645     51264521      49   Female         Non-resident  White
      2   55514647     51264523                                          None
      3   55514648     51264524      28   Female             Resident  White
      4   55514649     51264525      16     Male             Resident  White

            age_group               ethnicity              victim_type
      0  Age in Years  Not Hispanic or Latino  Law Enforcement Officer
      1  Age in Years                 Unknown               Individual
      2          None                    None           Society/Public
      3  Age in Years                 Unknown               Individual
      4  Age in Years                 Unknown               Individual
```

```
[98]: len(df_victim)
```

```
[98]: 3229640
```

```
[99]: q='SELECT * from weapon_main'
      df_weapon=table_query(q, cur)
      with open('data/pickled_dataframes/weapon.pickle', 'wb') as f:
          pickle.dump(df_weapon, f)
```

```
[100]: with open('data/pickled_dataframes/weapon.pickle', 'rb') as f:
           df_weapon=pickle.load(f)
       df_weapon.head()
```

```
[100]:    offense_id                 weapon
       0    53563151  Non-automatic firearm
       1    53558280  Non-automatic firearm
       2    53563153  Non-automatic firearm
       3    53579810  Non-automatic firearm
       4    53572975  Non-automatic firearm
```

```
[101]: len(df_weapon)
```

```
[101]: 551049
```

```
[102]: q='SELECT * from bias_main'
       df_bias=table_query(q, cur)
       with open('data/pickled_dataframes/bias.pickle', 'wb') as f:
           pickle.dump(df_bias, f)
```

[103]:

```
with open('data/pickled_dataframes/bias.pickle', 'rb') as f:
    df_bias=pickle.load(f)
df_bias.head()
```

[103]:

```
   offense_id bias_name
0    53563151      None
1    53563402      None
2    53558278      None
3    53558279      None
4    53563403      None
```

[104]:

```
len(df_bias)
```

[104]:

```
3201158
```

[105]:

```
q='SELECT * from victim_offender_rel'
df_rel=table_query(q, cur)
with open('data/pickled_dataframes/relationship.pickle', 'wb') as f:
    pickle.dump(df_rel, f)
```

[106]:

```
with open('data/pickled_dataframes/relationship.pickle', 'rb') as f:
    df_rel=pickle.load(f)
df_rel.head()
```

[106]:

```
   victim_id  offender_id                relationship_name
0   55514644     57702592       Victim was Otherwise Known
1   55514649     57702597             Victim Was Stepchild
2   55514652     57702601                Victim Was Spouse
3   55514653     57702602  Victim Was Boyfriend/Girlfriend
4   55514655     57702604                 Victim Was Child
```

[107]:

```
len(df_rel)
```

[107]:

```
794157
```

[108]:

```
cur.close()
conn.commit()
conn.close()
```

It takes 13 minutes to run this notebook from top to bottom

The next step is pre-processing data in DataFrames and EDA in scrub, part 2 notebook

[ ]:

```
```