**This document contains high level steps taken during completion of the assignment along with the outputs**

**Assignment 1**

Both success criteria were met for this assignment. Attached is the screenshot below

A screenshot of a computer program

Description automatically generated

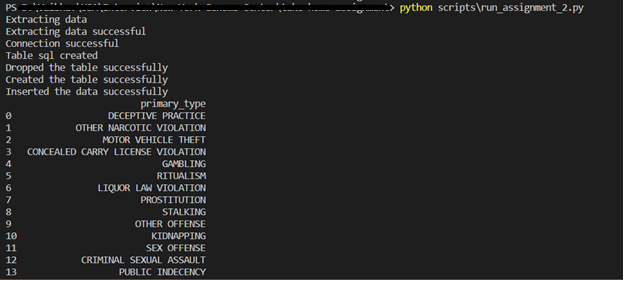
Output files were generated in the .outputs folder

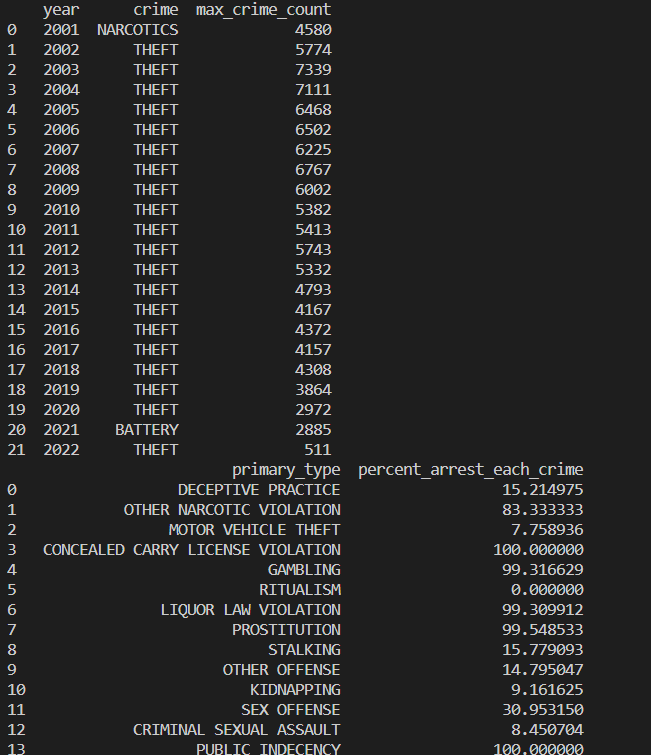
A screenshot of a computer

Description automatically generated

**Assignment 2**

To run the assignment, run the following statement -> **python scripts\run\_assignment\_2.py**

****

****

**A screenshot of a computer

Description automatically generated**

Below is the screenshot of the data-model/data-schema for the crimes table. Only the columns that were required for answering the 5 sql queries were selected instead of bringing all the columns. The data model is used to build the table dynamically based on the data types/constraints/primary keys.

**A screen shot of a computer program

Description automatically generated**

All the SQL files are present in the sql folder. They answer all the questions below in their order:

* select\_query\_1: - What is the list of all the types of committed crimes (primary\_type) in the dataset?
* select\_query\_2: - For any given year, what type of crime (primary\_type) was the most frequently committed?
* select\_query\_3: - What percentage of each type of crime (primary\_type) ended in arrest (arrest == true)?
* select\_query\_4: - is the frequency of each crime type (primary\_type) year over year? Imagine we want to generate datapoints to graph the number of occurrences of each crime type over every year in the dataset.
* select\_query\_5: - For any beat, district, ward or community (only one will be provided at a time), retrieve all the unique keys of each crime incident ordered by date.

**A screen shot of a computer

Description automatically generated**

**Assignment 3**

All the success criteria were met for this assignment:

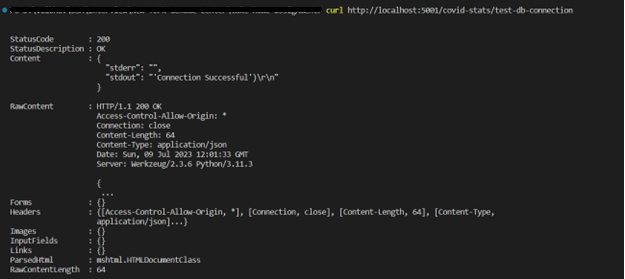
* Records from the covid\_state\_stats table are served via the new endpoint.
* Records are served in json format.
* Datetime fields are in the ISO 8601 format.
* Results are paginated

Once you have the connection established and have executed python -m api, the following endpoint <http://localhost:5001/covid-stats/get-covid-state-stats> will provide the covid data.

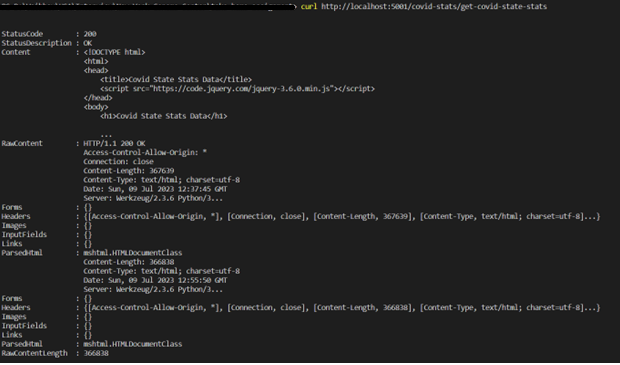
Attached are some of the screenshots on how the executions look like.

****

* **test-db-connection end point**

****

* **get-covid-state-stats end point**

****

* **get-covid-state-stats end point data looks like this in the json format. Each page contain 1000 json formatted data by default.**

**A screenshot of a computer

Description automatically generated**

**Following shows the pagination & end of the 1st page.**

**A screenshot of a computer program

Description automatically generated**