mod5_final_project

June 27, 2025

Assignment: Notebook for Graded Assessment

1 Introduction

Using this Python notebook you will:

1. Understand three Chicago datasets
2. Load the three datasets into three tables in a SQLIte database
3. Execute SQL queries to answer assignment questions

1.1 Understand the datasets

To complete the assignment problems in this notebook you will be using three datasets that are available on the city of Chicago's Data Portal:

1. Socioeconomic Indicators in Chicago
2. Chicago Public Schools
3. Chicago Crime Data

1.1.1 1. Socioeconomic Indicators in Chicago

This dataset contains a selection of six socioeconomic indicators of public health significance and a "hardship index," for each Chicago community area, for the years 2008 – 2012.

A detailed description of this dataset and the original dataset can be obtained from the Chicago Data Portal at:

https://data.cityofchicago.org/Health-Human-Services/Census-Data-Selected-socioeconomic-indicators-in-C/kn9c-c2s2

1.1.2 2. Chicago Public Schools

This dataset shows all school level performance data used to create CPS School Report Cards for the 2011-2012 school year. This dataset is provided by the city of Chicago's Data Portal.

A detailed description of this dataset and the original dataset can be obtained from the Chicago Data Portal at:

https://data.cityofchicago.org/Education/Chicago-Public-Schools-Progress-Report-Cards-2011-/9xs2-f89t

1.1.3 3. Chicago Crime Data

This dataset reflects reported incidents of crime (with the exception of murders where data exists for each victim) that occurred in the City of Chicago from 2001 to present, minus the most recent seven days.

A detailed description of this dataset and the original dataset can be obtained from the Chicago Data Portal at:

https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2

1.1.4 Download the datasets

This assignment requires you to have these three tables populated with a subset of the whole datasets.

In many cases the dataset to be analyzed is available as a .CSV (comma separated values) file, perhaps on the internet.

Use the links below to read the data files using the Pandas library.

- Chicago Census Data

https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/FinalModule_Coursera_V5/data/ChicagoCensusData.csv?utm_medium=Exinflu
SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDB0201ENSkillsNetwork20127838-2021-01-01

- Chicago Public Schools

https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/FinalModule_Coursera_V5/data/ChicagoPublicSchools.csv?utm_medium=Exin
SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDB0201ENSkillsNetwork20127838-2021-01-01

- Chicago Crime Data

https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/FinalModule_Coursera_V5/data/ChicagoCrimeData.csv?utm_medium=Exinflu
SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkDB0201ENSkillsNetwork20127838-2021-01-01

NOTE: Ensure you use the datasets available on the links above instead of directly from the Chicago Data Portal. The versions linked here are subsets of the original datasets and have some of the column names modified to be more database friendly which will make it easier to complete this assignment.

Execute the below code cell to install the required libraries

[1]:

```
!pip install pandas
!pip install ipython-sql prettytable

import prettytable
```

```python
prettytable.DEFAULT = 'DEFAULT'
```

```
Collecting pandas
  Downloading
pandas-2.3.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata
(91 kB)
Collecting numpy>=1.26.0 (from pandas)
  Downloading numpy-2.3.1-cp312-cp312-manylinux_2_28_x86_64.whl.metadata (62 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in
/opt/conda/lib/python3.12/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.12/site-
packages (from pandas) (2024.2)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.12/site-
packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Downloading
pandas-2.3.0-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (12.0
MB)
                         12.0/12.0 MB
182.2 MB/s eta 0:00:00
Downloading numpy-2.3.1-cp312-cp312-manylinux_2_28_x86_64.whl (16.6 MB)
                         16.6/16.6 MB
161.0 MB/s eta 0:00:00
Downloading tzdata-2025.2-py2.py3-none-any.whl (347 kB)
Installing collected packages: tzdata, numpy, pandas
Successfully installed numpy-2.3.1 pandas-2.3.0 tzdata-2025.2
Collecting ipython-sql
  Downloading ipython_sql-0.5.0-py3-none-any.whl.metadata (17 kB)
Collecting prettytable
  Downloading prettytable-3.16.0-py3-none-any.whl.metadata (33 kB)
Requirement already satisfied: ipython in /opt/conda/lib/python3.12/site-
packages (from ipython-sql) (8.31.0)
Requirement already satisfied: sqlalchemy>=2.0 in
/opt/conda/lib/python3.12/site-packages (from ipython-sql) (2.0.37)
Collecting sqlparse (from ipython-sql)
  Downloading sqlparse-0.5.3-py3-none-any.whl.metadata (3.9 kB)
Requirement already satisfied: six in /opt/conda/lib/python3.12/site-packages
(from ipython-sql) (1.17.0)
Requirement already satisfied: ipython-genutils in
/opt/conda/lib/python3.12/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.12/site-
packages (from prettytable) (0.2.13)
Requirement already satisfied: greenlet!=0.4.17 in
/opt/conda/lib/python3.12/site-packages (from sqlalchemy>=2.0->ipython-sql)
(3.1.1)
Requirement already satisfied: typing-extensions>=4.6.0 in
/opt/conda/lib/python3.12/site-packages (from sqlalchemy>=2.0->ipython-sql)
```

```
(4.12.2)
Requirement already satisfied: decorator in /opt/conda/lib/python3.12/site-
packages (from ipython->ipython-sql) (5.1.1)
Requirement already satisfied: jedi>=0.16 in /opt/conda/lib/python3.12/site-
packages (from ipython->ipython-sql) (0.19.2)
Requirement already satisfied: matplotlib-inline in
/opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (0.1.7)
Requirement already satisfied: pexpect>4.3 in /opt/conda/lib/python3.12/site-
packages (from ipython->ipython-sql) (4.9.0)
Requirement already satisfied: prompt_toolkit<3.1.0,>=3.0.41 in
/opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (3.0.50)
Requirement already satisfied: pygments>=2.4.0 in
/opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (2.19.1)
Requirement already satisfied: stack_data in /opt/conda/lib/python3.12/site-
packages (from ipython->ipython-sql) (0.6.3)
Requirement already satisfied: traitlets>=5.13.0 in
/opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (5.14.3)
Requirement already satisfied: parso<0.9.0,>=0.8.4 in
/opt/conda/lib/python3.12/site-packages (from jedi>=0.16->ipython->ipython-sql)
(0.8.4)
Requirement already satisfied: ptyprocess>=0.5 in
/opt/conda/lib/python3.12/site-packages (from pexpect>4.3->ipython->ipython-sql)
(0.7.0)
Requirement already satisfied: executing>=1.2.0 in
/opt/conda/lib/python3.12/site-packages (from stack_data->ipython->ipython-sql)
(2.1.0)
Requirement already satisfied: asttokens>=2.1.0 in
/opt/conda/lib/python3.12/site-packages (from stack_data->ipython->ipython-sql)
(3.0.0)
Requirement already satisfied: pure_eval in /opt/conda/lib/python3.12/site-
packages (from stack_data->ipython->ipython-sql) (0.2.3)
Downloading ipython_sql-0.5.0-py3-none-any.whl (20 kB)
Downloading prettytable-3.16.0-py3-none-any.whl (33 kB)
Downloading sqlparse-0.5.3-py3-none-any.whl (44 kB)
Installing collected packages: sqlparse, prettytable, ipython-sql
Successfully installed ipython-sql-0.5.0 prettytable-3.16.0 sqlparse-0.5.3
```

1.1.5 Store the datasets in database tables

To analyze the data using SQL, it first needs to be loaded into SQLite DB. We will create three tables in as under:

1. **CENSUS_DATA**
2. **CHICAGO_PUBLIC_SCHOOLS**
3. **CHICAGO_CRIME_DATA**

Load the `pandas` and `sqlite3` libraries and establish a connection to `FinalDB.db`

[2]:

```
import pandas, csv, sqlite3

con = sqlite3.connect('FinalDB.db')
```

Load the SQL magic module

[3]:

```
%load_ext sql
%sql sqlite:///FinalDB.db
```

Use `Pandas` to load the data available in the links above to dataframes. Use these dataframes to load data on to the database `FinalDB.db` as required tables.

[7]:

```
df_ps = pandas.read_csv('ChicagoPublicSchools.csv')
df_cen = pandas.read_csv('ChicagoCensusData.csv')
df_cr = pandas.read_csv('ChicagoCrimeData.csv')

df_ps.to_sql("Chicago_Public_Schools", con, if_exists='replace', index=False,␣
 ↪method='multi')
df_cen.to_sql("Chicago_Census", con, if_exists='replace', index=False,␣
 ↪method='multi')
df_cr.to_sql("Chicago_Crime", con, if_exists='replace', index=False,␣
 ↪method='multi')
```

[7]:

```
533
```

Establish a connection between SQL magic module and the database `FinalDB.db`

[ ]:

```
# DONE ABOVE UNDER `Load the SQL magic module`
```

You can now proceed to the the following questions. Please note that a graded assignment will follow this lab and there will be a question on each of the problems stated below. It can be from the answer you received or the code you write for this problem. Therefore, please keep a note of both your codes as well as the response you generate.

1.2 Problems

Now write and execute SQL queries to solve assignment problems

1.2.1 Problem 1

Find the total number of crimes recorded in the CRIME table.

[8]:

```
%%sql
SELECT count(*) AS Total_Recorded_Crimes
FROM Chicago_Crime;
```

```
 * sqlite:///FinalDB.db
Done.
```

[8]:

```
[(533,)]
```

1.2.2 Problem 2

List community area names and numbers with per capita income less than 11000.

[10]:
```
%%sql

SELECT COMMUNITY_AREA_NUMBER, COMMUNITY_AREA_NAME, PER_CAPITA_INCOME
FROM Chicago_Census
WHERE PER_CAPITA_INCOME < 11000
```

```
 * sqlite:///FinalDB.db
Done.
```

[10]:
```
[(26.0, 'West Garfield Park', 10934),
 (30.0, 'South Lawndale', 10402),
 (37.0, 'Fuller Park', 10432),
 (54.0, 'Riverdale', 8201)]
```

1.2.3 Problem 3

List all case numbers for crimes involving minors?(children are not considered minors for the purposes of crime analysis)

[11]:
```
%%sql
SELECT CASE_NUMBER, PRIMARY_TYPE, DESCRIPTION
FROM Chicago_Crime
WHERE UPPER(PRIMARY_TYPE) LIKE '%MINOR%' OR UPPER(DESCRIPTION) LIKE '%MINOR%';
```

```
 * sqlite:///FinalDB.db
Done.
```

[11]:
```
[('HL266884', 'LIQUOR LAW VIOLATION', 'SELL/GIVE/DEL LIQUOR TO MINOR'),
 ('HK238408', 'LIQUOR LAW VIOLATION', 'ILLEGAL CONSUMPTION BY MINOR')]
```

1.2.4 Problem 4

List all kidnapping crimes involving a child?

[14]:
```
%%sql
SELECT CASE_NUMBER, PRIMARY_TYPE, DESCRIPTION
FROM Chicago_Crime
WHERE UPPER(PRIMARY_TYPE) = "KIDNAPPING" AND UPPER(DESCRIPTION) LIKE '%CHILD%';
```

```
 * sqlite:///FinalDB.db
Done.
```

[14]:
```
[('HN144152', 'KIDNAPPING', 'CHILD ABDUCTION/STRANGER')]
```

1.2.5 Problem 5

List the kind of crimes that were recorded at schools. (No repetitions)

[17]:
```
%%sql
SELECT PRIMARY_TYPE, COUNT(*) AS COUNT_OF_CRIMES_OCCURRED
```

```
FROM Chicago_Crime
WHERE UPPER(LOCATION_DESCRIPTION) LIKE '%SCHOOL%'
GROUP BY PRIMARY_TYPE
ORDER BY COUNT_OF_CRIMES_OCCURRED DESC;
```

```
 * sqlite:///FinalDB.db
Done.
```

```
[17]: [('BATTERY', 5),
       ('PUBLIC PEACE VIOLATION', 2),
       ('NARCOTICS', 2),
       ('CRIMINAL TRESPASS', 1),
       ('CRIMINAL DAMAGE', 1),
       ('ASSAULT', 1)]
```

1.2.6 Problem 6

List the type of schools along with the average safety score for each type.

```
[21]: %%sql
      SELECT `Elementary, Middle, or High School` AS TYPE_OF_SCHOOL,␣
        ↪AVG(SAFETY_SCORE) AS AVERAGE_SAFETY_SCORE
      FROM Chicago_Public_Schools
      GROUP BY TYPE_OF_SCHOOL
      ORDER BY AVERAGE_SAFETY_SCORE DESC;
```

```
 * sqlite:///FinalDB.db
Done.
```

```
[21]: [('HS', 49.62352941176471), ('ES', 49.52038369304557), ('MS', 48.0)]
```

1.2.7 Problem 7

List 5 community areas with highest % of households below poverty line

```
[24]: %%sql
      SELECT CAST(COMMUNITY_AREA_NUMBER AS INT) AS COMMUNITY_AREA_NUMBER,␣
        ↪COMMUNITY_AREA_NAME, PERCENT_HOUSEHOLDS_BELOW_POVERTY
      FROM Chicago_Census
      ORDER BY PERCENT_HOUSEHOlDS_BELOW_POVERTY DESC LIMIT 5;
```

```
 * sqlite:///FinalDB.db
Done.
```

```
[24]: [(54, 'Riverdale', 56.5),
       (37, 'Fuller Park', 51.2),
       (68, 'Englewood', 46.6),
       (29, 'North Lawndale', 43.1),
       (27, 'East Garfield Park', 42.4)]
```

1.2.8 Problem 8

Which community area is most crime prone? Display the coumminty area number only.

[27]:
```
%%sql
SELECT CAST(COMMUNITY_AREA_NUMBER AS INT) AS COMMUNITY_AREA_NUMBER
FROM (SELECT COMMUNITY_AREA_NUMBER, COUNT(*)
    AS CRIME_TOTAL
    FROM Chicago_Crime
    GROUP BY COMMUNITY_AREA_NUMBER
    ORDER BY CRIME_TOTAL DESC nulls last LIMIT 1);
```

```
 * sqlite:///FinalDB.db
Done.
```

[27]:
```
[(25,)]
```

Double-click **here** for a hint

1.2.9 Problem 9

Use a sub-query to find the name of the community area with highest hardship index

[28]:
```
%%sql
SELECT COMMUNITY_AREA_NAME
FROM Chicago_Census
WHERE HARDSHIP_INDEX = (SELECT MAX(HARDSHIP_INDEX) FROM Chicago_Census);
```

```
 * sqlite:///FinalDB.db
Done.
```

[28]:
```
[('Riverdale',)]
```

1.2.10 Problem 10

Use a sub-query to determine the Community Area Name with most number of crimes?

[31]:
```
%%sql
SELECT COMMUNITY_AREA_NAME
FROM Chicago_Census
WHERE COMMUNITY_AREA_NUMBER = (SELECT COMMUNITY_AREA_NUMBER FROM (SELECT␣
 ↪COMMUNITY_AREA_NUMBER, COUNT(*) AS CRIME_TOTAL
    FROM Chicago_Crime
    GROUP BY COMMUNITY_AREA_NUMBER
    ORDER BY CRIME_TOTAL DESC nulls last LIMIT 1));
```

```
 * sqlite:///FinalDB.db
Done.
```

[31]:
```
[('Austin',)]
```

1.3 Author(s)

Hima Vasudevan

Rav Ahuja

Ramesh Sannreddy

1.4 Contribtuor(s)

Malika Singla

Abhishek Gagneja

##