



## Assignment: Notebook for Graded Assessment

### 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

### 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. 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>

#### 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>

### 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>

## 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/IBMDriverSkillsNetwork-DB0201EN-SkillsNetwork/labs/FinalModule\\_Coursera\\_V5/data/ChicagoCensusData.csv?utm\\_medium=Exinfluencer&utm\\_source=Exinfluencer&utm\\_content=000026UJ&utm\\_term=100SkillsNetwork-Channel-SkillsNetworkCoursesIBMDriverSkillsNetworkDB0201ENSkillNetwork20127838-2021-01-01](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDriverSkillsNetwork-DB0201EN-SkillsNetwork/labs/FinalModule_Coursera_V5/data/ChicagoCensusData.csv?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=100SkillsNetwork-Channel-SkillsNetworkCoursesIBMDriverSkillsNetworkDB0201ENSkillNetwork20127838-2021-01-01)

- Chicago Public Schools

[https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDriverSkillsNetwork-DB0201EN-SkillsNetwork/labs/FinalModule\\_Coursera\\_V5/data/ChicagoPublicSchools.csv?utm\\_medium=Exinfluencer&utm\\_source=Exinfluencer&utm\\_content=000026UJ&utm\\_term=100SkillsNetwork-Channel-SkillsNetworkCoursesIBMDriverSkillsNetworkDB0201ENSkillNetwork20127838-2021-01-01](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDriverSkillsNetwork-DB0201EN-SkillsNetwork/labs/FinalModule_Coursera_V5/data/ChicagoPublicSchools.csv?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=100SkillsNetwork-Channel-SkillsNetworkCoursesIBMDriverSkillsNetworkDB0201ENSkillNetwork20127838-2021-01-01)

- Chicago Crime Data

[https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDriverSkillsNetwork-DB0201EN-SkillsNetwork/labs/FinalModule\\_Coursera\\_V5/data/ChicagoCrimeData.csv?utm\\_medium=Exinfluencer&utm\\_source=Exinfluencer&utm\\_content=000026UJ&utm\\_term=100SkillsNetwork-Channel-SkillsNetworkCoursesIBMDriverSkillsNetworkDB0201ENSkillNetwork20127838-2021-01-01](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDriverSkillsNetwork-DB0201EN-SkillsNetwork/labs/FinalModule_Coursera_V5/data/ChicagoCrimeData.csv?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=100SkillsNetwork-Channel-SkillsNetworkCoursesIBMDriverSkillsNetworkDB0201ENSkillNetwork20127838-2021-01-01)

**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 avoid prettytable default error.

```
In [2]: !pip install ipython-sql prettytable  
  
import prettytable  
  
prettytable.DEFAULT = 'DEFAULT'
```

```
Collecting ipython-sql
  Downloading ipython_sql-0.5.0-py3-none-any.whl.metadata (17 kB)
Collecting prettytable
  Downloading prettytable-3.14.0-py3-none-any.whl.metadata (30 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-3.14.0-py3-none-any.whl (31 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-3.14.0 sqlparse-0.5.3
```

## 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`

```
In [3]: !pip install pandas
import pandas as pd
import sqlite3
```

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

Collecting pandas

  Downloading pandas-2.2.3-cp312-cp312-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl.metadata (89 kB)

Collecting numpy>=1.26.0 (from pandas)

  Downloading numpy-2.2.2-cp312-cp312-manylinux\_2\_17\_x86\_64.manylinux2014\_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-2025.1-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-2.2.3-cp312-cp312-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (12.7 MB)

                        12.7/12.7 MB 173.5 MB/s eta 0:00:00

  Downloading numpy-2.2.2-cp312-cp312-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (16.1 MB)

                        16.1/16.1 MB 201.3 MB/s eta 0:00:00

  Downloading tzdata-2025.1-py2.py3-none-any.whl (346 kB)

Installing collected packages: tzdata, numpy, pandas

Successfully installed numpy-2.2.2 pandas-2.2.3 tzdata-2025.1

Load the SQL magic module

```
In [4]: %load_ext sql
```

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.

```
In [ ]: Chicago_Census_Data = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-stora
Chicago_Public_Schools = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-st
Chicago_Crime_Data = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storag

Chicago_Census_Data.to_sql('CENSUS_DATA', conn)
Chicago_Public_Schools.to_sql('CHICAGO_PUBLIC_SCHOOLS', conn)
Chicago_Crime_Data.to_sql('CHICAGO_CRIME_DATA', conn)
```

```
In [6]: type(Chicago_Census_Data), type(Chicago_Public_Schools), type(Chicago_Crime_Data)
```

```
Out[6]: (pandas.core.frame.DataFrame,
         pandas.core.frame.DataFrame,
         pandas.core.frame.DataFrame)
```

```
In [6]: Chicago_Census_Data.head(3)
```

	COMMUNITY_AREA_NUMBER	COMMUNITY_AREA_NAME	PERCENT_OF_HOUSING_CROWD
<b>0</b>	1.0	Rogers Park	
<b>1</b>	2.0	West Ridge	
<b>2</b>	3.0	Uptown	

```
In [7]: Chicago_Public_Schools.head(3)
```

	School_ID	NAME_OF SCHOOL	Elementary, Middle, or High School	Street_Address	City	State	ZIP_Code	F
<b>0</b>	610038	Abraham Lincoln Elementary School	ES	615 W Kemper Pl	Chicago	IL	60614	
<b>1</b>	610281	Adam Clayton Powell Paideia Community Academy ...	ES	7511 S South Shore Dr	Chicago	IL	60649	
<b>2</b>	610185	Adlai E Stevenson Elementary School	ES	8010 S Kostner Ave	Chicago	IL	60652	

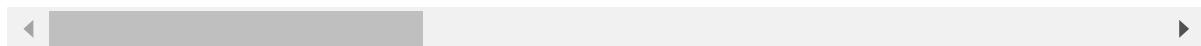
3 rows × 78 columns

```
In [8]: Chicago_Crime_Data.head(3)
```

Out[8]:

	ID	CASE_NUMBER	DATE	BLOCK	IUCR	PRIMARY_TYPE	DESCRIPTION	LOCATION_DESCRIPTION	ARREST	DOMESTIC	BEAT	DISTRICT	WARD	COMMUNITY_AREA_NUMBER	FBICODE	X_COORDINATE	Y_COORDINATE	YEAR	LATITUDE	LONGITUDE	LOCATION
0	3512276	HK587712	2004-08-28	047XX S KEDZIE AVE	890	THEFT	FROM BUILDING	S													
1	3406613	HK456306	2004-06-26	009XX N CENTRAL PARK AVE	820	THEFT	\$500 AND UNDER														
2	8002131	HT233595	2011-04-04	043XX S WABASH AVE	820	THEFT	\$500 AND UNDER	F													

3 rows × 21 columns



In [13]: `Chicago_Crime_Data.columns.tolist()`

Out[13]:

```
[ 'ID',
  'CASE_NUMBER',
  'DATE',
  'BLOCK',
  'IUCR',
  'PRIMARY_TYPE',
  'DESCRIPTION',
  'LOCATION_DESCRIPTION',
  'ARREST',
  'DOMESTIC',
  'BEAT',
  'DISTRICT',
  'WARD',
  'COMMUNITY_AREA_NUMBER',
  'FBICODE',
  'X_COORDINATE',
  'Y_COORDINATE',
  'YEAR',
  'LATITUDE',
  'LONGITUDE',
  'LOCATION']
```

In [55]: `Chicago_Crime_Data.DESCRIPTION[520:]`

```
Out[55]: 520          CHILD ABDUCTION/STRANGER
      521          INTIMIDATION
      522          CYBERSTALKING
      523          OBSCENE MATTER
      524          OTHER
      525          LICENSED PREMISE
      526          LOST PASSPORT
      527          INTOXICATING COMPOUNDS
      528          INVOLUNTARY SERVITUDE
      529          FOID - REVOCATION
      530  AGG RIT MUT: HANDS/FIST/FEET SERIOUS INJURY
      531          NOTIFICATION OF CIVIL NO CONTACT ORDER
      532          DOMESTIC VIOLENCE
Name: DESCRIPTION, dtype: object
```

In [83]: Chicago\_Crime\_Data.PRIMARY\_TYPE[520:]

```
Out[83]: 520          KIDNAPPING
      521          INTIMIDATION
      522          STALKING
      523          OBSCENITY
      524  CONCEALED CARRY LICENSE VIOLATION
      525          PUBLIC INDECENCY
      526          NON-CRIMINAL
      527          OTHER NARCOTIC VIOLATION
      528          HUMAN TRAFFICKING
      529          NON - CRIMINAL
      530          RITUALISM
      531          NON-CRIMINAL (SUBJECT SPECIFIED)
      532          DOMESTIC VIOLENCE
Name: PRIMARY_TYPE, dtype: object
```

In [95]: #Chicago\_Public\_Schools.columns.tolist()

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

In [7]: %sql sqlite:///FinalDB.db

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.

## Problems

Now write and execute SQL queries to solve assignment problems

### Problem 1

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

```
In [9]: %sql SELECT COUNT(*) FROM CHICAGO_CRIME_DATA
* sqlite:///FinalDB.db
Done.

Out[9]: COUNT(*)
533
```

## Problem 2

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

```
In [10]: %sql SELECT COMMUNITY_AREA_NAME, COMMUNITY_AREA_NUMBER, PER_CAPITA_INCOME \
FROM CENSUS_DATA \
WHERE PER_CAPITA_INCOME < 11000
* sqlite:///FinalDB.db
Done.

Out[10]: COMMUNITY_AREA_NAME  COMMUNITY_AREA_NUMBER  PER_CAPITA_INCOME
West Garfield Park          26.0                  10934
South Lawndale               30.0                  10402
Fuller Park                 37.0                  10432
Riverdale                    54.0                  8201
```

## Problem 3

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

```
In [88]: %sql SELECT * FROM CHICAGO_CRIME_DATA WHERE DESCRIPTION IN ('CHILD', 'MINOR', 'JUVE
* sqlite:///FinalDB.db
Done.

Out[88]: index      ID  CASE_NUMBER  DATE   BLOCK  IUCR  PRIMARY_TYPE      DESCRIPTIC
520  5276766      HN144152  2007-01-26  050XX
                                W
                                VAN
                                BUREN
                                ST
                                KIDNAPPING
                                ABDUCTION/STRANG
                                CHII
```

## Problem 4

List all kidnapping crimes involving a child?

```
In [90]: %sql SELECT * FROM CHICAGO_CRIME_DATA WHERE PRIMARY_TYPE = 'KIDNAPPING' and DESCRIPTIC
```

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

Out[90]:	index	ID	CASE_NUMBER	DATE	BLOCK	IUCR	PRIMARY_TYPE	DESCRIPTION
	520	5276766	HN144152	2007-01-26	050XX W VAN BUREN ST	1792	KIDNAPPING	CHII ABDUCTION/STRANG

◀ ▶

```
In [35]: # %sql SELECT * FROM CHICAGO_CRIME_DATA WHERE PRIMARY_TYPE = 'KIDNAPPING' and DESCRIPTI
```

## Problem 5

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

```
In [92]: %sql SELECT DISTINCT(PRIMARY_TYPE) FROM CHICAGO_CRIME_DATA WHERE LOCATION_DESCRIPTION LIKE '%SCHOOL%'
```

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

```
Out[92]: PRIMARY_TYPE
```

BATTERY
CRIMINAL DAMAGE
NARCOTICS
ASSAULT
CRIMINAL TRESPASS
PUBLIC PEACE VIOLATION

## Problem 6

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

```
In [104...]: %sql SELECT DISTINCT(`Elementary, Middle, or High School`), AVG(SAFETY_SCORE) FROM CHICAGO_SCHOOL_SAFETY
```

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

```
Out[104...]: Elementary, Middle, or High School  AVG(SAFETY_SCORE)
```

ES	49.52038369304557
HS	49.62352941176471
MS	48.0

## Problem 7

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

```
In [105]: %%sql SELECT COMMUNITY_AREA_NAME, PERCENT_HOUSEHOLDS_BELOW_POVERTY FROM CENSUS_DATA
* sqlite:///FinalDB.db
Done.
```

COMMUNITY_AREA_NAME	PERCENT_HOUSEHOLDS_BELOW_POVERTY
Riverdale	56.5
Fuller Park	51.2
Englewood	46.6
North Lawndale	43.1
East Garfield Park	42.4

## Problem 8

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

```
In [10]: %%sql SELECT COMMUNITY_AREA_NUMBER
        FROM CHICAGO_CRIME_DATA
        GROUP BY COMMUNITY_AREA_NUMBER
        ORDER BY COUNT(*) DESC
        LIMIT 1;
```

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

```
Out[10]: COMMUNITY_AREA_NUMBER
```

25.0
------

Double-click **here** for a hint

```
In [11]: # GROUP BY COMMUNITY_AREA_NUMBER: Agrupa los crímenes por número de área comunitaria
# ORDER BY COUNT(*) DESC: Cuenta la cantidad de crímenes en cada área y ordena de mayor a menor
# LIMIT 1: Devuelve solo la comunidad con más crímenes.
```

## Problem 9

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

```
In [18]: %%sql SELECT COMMUNITY_AREA_NAME
        FROM CENSUS_DATA
        WHERE HARDSHIP_INDEX = (SELECT MAX(HARDSHIP_INDEX) FROM CENSUS_DATA);
```

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

```
Out[18]: COMMUNITY_AREA_NAME
```

Riverdale
-----------

```
In [34]: # %%sql SELECT COMMUNITY_AREA_NAME
#           FROM CENSUS_DATA
#           WHERE HARSHIP_INDEX IN (SELECT MAX(HARDSHIP_INDEX) FROM CENSUS_DATA);
```

## Problem 10

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

```
In [33]: %%sql SELECT COMMUNITY_AREA_NAME
           FROM CENSUS_DATA
           WHERE COMMUNITY_AREA_NUMBER = (SELECT COMMUNITY_AREA_NUMBER
           FROM CHICAGO_CRIME_DATA
           GROUP BY COMMUNITY_AREA_NUMBER
           ORDER BY COUNT(*) DESC
           LIMIT 1;)
```

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

Out[33]: **COMMUNITY\_AREA\_NAME**

Austin

## Author(s)

Hima Vasudevan

Rav Ahuja

Ramesh Sannreddy

## Contributor(s)

Malika Singla

Abhishek Gagneja

```
<!-- ## Change log <table> Date Version Changed by Change Description 2023-10-18 2.6
Abhishek Gagneja Modified instruction set 2022-03-04 2.5 Lakshmi Holla Changed
markdown. 2021-05-19 2.4 Lakshmi Holla Updated the question 2021-04-30 2.3 Malika
Singla Updated the libraries 2021-01-15 2.2 Rav Ahuja Removed problem 11 and fixed
changelog 2020-11-25 2.1 Ramesh Sannreddy Updated the problem statements, and
datasets 2020-09-05 2.0 Malika Singla Moved lab to course repo in GitLab 2018-07-18 1.0
Rav Ahuja Several updates including loading instructions 2018-05-04 0.1 Hima Vasudevan
Created initial version
-->
```

© IBM Corporation 2023. All rights reserved.

