



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

- 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

- 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-SkillsNetworkCoursesIBMDriverSkillsNetworkDB0201ENSkillsNetwork20127838-2021-01-01

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.

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 [5]: import csv, sqlite3  
  
con = sqlite3.connect("FinalDB.db")  
cur = con.cursor()
```

```
In [6]: !pip install pandas
```

```
Requirement already satisfied: pandas in /opt/conda/lib/python3.11/site-packages (2.2.3)  
Requirement already satisfied: numpy>=1.23.2 in /opt/conda/lib/python3.11/site-packages (from pandas) (2.1.2)  
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/conda/lib/python3.11/site-packages (from pandas) (2.9.0)  
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.11/site-packages (from pandas) (2024.1)  
Requirement already satisfied: tzdata>=2022.7 in /opt/conda/lib/python3.11/site-packages (from pandas) (2024.2)  
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.11/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

Load the SQL magic module

```
In [7]: !pip install ipython-sql  
%load_ext sql
```

```
Requirement already satisfied: ipython-sql in /opt/conda/lib/python3.11/site-packages (0.5.0)  
Requirement already satisfied: prettytable in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (3.11.0)  
Requirement already satisfied: ipython in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (8.22.2)  
Requirement already satisfied: sqlalchemy>=2.0 in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (2.0.30)  
Requirement already satisfied: sqlparse in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (0.5.1)  
Requirement already satisfied: six in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (1.16.0)  
Requirement already satisfied: ipython-genutils in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (0.2.0)  
Requirement already satisfied: typing-extensions>=4.6.0 in /opt/conda/lib/python3.11/site-packages (from sqlalchemy>=2.0->ipython-sql) (4.11.0)  
Requirement already satisfied: greenlet!=0.4.17 in /opt/conda/lib/python3.11/site-packages (from sqlalchemy>=2.0->ipython-sql) (3.0.3)  
Requirement already satisfied: decorator in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (5.1.1)  
Requirement already satisfied: jedi>=0.16 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (0.19.1)  
Requirement already satisfied: matplotlib-inline in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (0.1.7)  
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (3.0.42)  
Requirement already satisfied: pygments>=2.4.0 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (2.18.0)  
Requirement already satisfied: stack-data in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (0.6.2)  
Requirement already satisfied: traitlets>=5.13.0 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (5.14.3)  
Requirement already satisfied: pexpect>4.3 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (4.9.0)  
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.11/site-packages (from prettytable->ipython-sql) (0.2.13)  
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /opt/conda/lib/python3.11/site-packages (from jedi>=0.16->ipython->ipython-sql) (0.8.4)  
Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/lib/python3.11/site-packages (from pexpect>4.3->ipython->ipython-sql) (0.7.0)  
Requirement already satisfied: executing>=1.2.0 in /opt/conda/lib/python3.11/site-packages (from stack-data->ipython->ipython-sql) (2.0.1)  
Requirement already satisfied: asttokens>=2.1.0 in /opt/conda/lib/python3.11/site-packages (from stack-data->ipython->ipython-sql) (2.4.1)  
Requirement already satisfied: pure-eval in /opt/conda/lib/python3.11/site-packages (from stack-data->ipython->ipython-sql) (0.2.2)  
The sql extension is already loaded. To reload it, use:  
%reload_ext sql
```

```
In [8]: %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.

```
In [9]: import pandas

df = pandas.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.
df.to_sql("CENSUS_DATA", con, if_exists='replace', index=False, method="multi")

df = pandas.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.
df.to_sql("CHICAGO_PUBLIC_SCHOOLS", con, if_exists='replace', index=False, method="

df = pandas.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.
df.to_sql("CHICAGO_CRIME_DATA", con, if_exists='replace', index=False, method="multi")
```

Out[9]: 533

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

In []:

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 [12]: %sql select count(*)from chicago_crime_data;
```

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

Out[12]: `count(*)`

533

Problem 2

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

```
In [14]: %%sql
```

```
select COMMUNITY_AREA_NAME , PER_CAPITA_INCOME from CENSUS_DATA where PER_CAPITA_IN
```

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

Out[14]: **COMMUNITY_AREA_NAME PER_CAPITA_INCOME**

West Garfield Park	10934
South Lawndale	10402
Fuller Park	10432
Riverdale	8201

Problem 3

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

In [15]:

```
%%sql
select CASE_NUMBER, PRIMARY_TYPE, DESCRIPTION from CHICAGO_CRIME_DATA where DESCRIPTI
* sqlite:///FinalDB.db
Done.
```

Out[15]:

CASE_NUMBER	PRIMARY_TYPE	DESCRIPTION
HL266884	LIQUOR LAW VIOLATION	SELL/GIVE/DEL LIQUOR TO MINOR
HK238408	LIQUOR LAW VIOLATION	ILLEGAL CONSUMPTION BY MINOR

Problem 4

List all kidnapping crimes involving a child?

In [16]:

```
%%sql
select * from CHICAGO_CRIME_DATA where PRIMARY_TYPE = 'KIDNAPPING' and DESCRIPTION
* sqlite:///FinalDB.db
Done.
```

Out[16]:

ID	CASE_NUMBER	DATE	BLOCK	IUCR	PRIMARY_TYPE	DESCRIPTION	LOC
5276766	HN144152	2007-01-26	050XX W VAN BUREN ST	1792	KIDNAPPING	CHILD ABDUCTION/STRANGER	

Problem 5

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

In [17]:

```
%%sql
select distinct(PRIMARY_TYPE) from CHICAGO_CRIME_DATA where LOCATION_DESCRIPTION li
* sqlite:///FinalDB.db
Done.
```

Out[17]:

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 [19]:

```
%%sql
select "Elementary, Middle, or High School", avg(safety_score) as average_safety_score
from CHICAGO_PUBLIC_SCHOOLS group by "Elementary, Middle, or High School"
* sqlite:///FinalDB.db
Done.
```

Out[19]:

Elementary, Middle, or High School average_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 [20]:

```
%%sql
select community_area_name, percent_households_below_poverty from CENSUS_DATA
order by percent_households_below_poverty desc nulls last limit 5;
* sqlite:///FinalDB.db
Done.
```

Out[20]:

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 [24]: %%sql
select COMMUNITY_AREA_NUMBER from census_data where community_area_name in( select
order by count(*) desc nulls last limit 1)

* sqlite:///FinalDB.db
Done.

Out[24]: COMMUNITY_AREA_NUMBER
```

42.0

Double-click **here** for a hint

Problem 9

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

```
In [25]: %%sql
select community_area_name from CENSUS_DATA where hardship_index = (select MAX(hard
* sqlite:///FinalDB.db
Done.

Out[25]: COMMUNITY_AREA_NAME
```

Riverdale

Problem 10

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

```
In [26]: %%sql
select COMMUNITY_AREA_NAME from CENSUS_DATA where COMMUNITY_AREA_NUMBER in
(select COMMUNITY_AREA_NUMBER from CHICAGO_CRIME_DATA group by COMMUNITY_AREA_NUMB
* sqlite:///FinalDB.db
Done.

Out[26]: 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 Sannareddy 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.