Assignment: Notebook for Peer Assignment

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. Click on the links below to download and save the datasets (.CSV files):

- Chicago Census Data
- Chicago Public Schools
- Chicago Crime Data

NOTE: Ensure you have downloaded the datasets using 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**

Let us now load the ipython-sql extension and establish a connection with the database

- Here you will be loading the csv files into the pandas Dataframe and then loading the data into the above mentioned sqlite tables.
- Next you will be connecting to the sqlite database **FinalDB**.

Refer to the previous lab for hints .

Hands-on Lab: Analyzing a real World Data Set

In [9]:
```
%load_ext sql
```

```
The sql extension is already loaded. To reload it, use:
  %reload_ext sql
```

In [11]:
```
import csv, sqlite3

con = sqlite3.connect("chicago.db")
cur = con.cursor()
!pip install -q pandas==1.1.5
```

In [12]:
```
%sql sqlite:///chicago.db
```

Out[12]:
```
'Connected: @chicago.db'
```

In [13]:
```
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_CRIME_DATA", con, if_exists='replace', index=False, method="mult

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

```
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/pandas/core/generic.p
y:2882: UserWarning: The spaces in these column names will not be changed. In pandas
versions < 0.14, spaces were converted to underscores.
  both result in 0.1234 being formatted as 0.12.
```

In [ ]:
```

```

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 [15]:
```
%sql SELECT COUNT(ID) FROM CHICAGO_CRIME_DATA
```

```
 * sqlite:///chicago.db
   sqlite:///socioeconomic.db
Done.
```

Out[15]:

COUNT(ID)
533

Problem 2

List community areas with per capita income less than 11000.

In [28]:
```
%sql SELECT COMMUNITY_AREA_NAME FROM CENSUS_DATA WHERE PER_CAPITA_INCOME < 11000
```

```
 * sqlite:///chicago.db
   sqlite:///socioeconomic.db
Done.
```

Out[28]:

COMMUNITY_AREA_NAME
West Garfield Park
South Lawndale
Fuller Park
Riverdale

Problem 3

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

In [29]:
```
%sql SELECT CASE_NUMBER FROM CHICAGO_CRIME_DATA WHERE DESCRIPTION LIKE '%MINOR%'
```

```
 * sqlite:///chicago.db
   sqlite:///socioeconomic.db
Done.
```

Out[29]:

CASE_NUMBER
HL266884
HK238408

Problem 4

List all kidnapping crimes involving a child?

In [32]:
```
%sql SELECT * FROM CHICAGO_CRIME_DATA WHERE DESCRIPTION LIKE '%CHILD%' AND PRIMARY_
```

```
 * sqlite:///chicago.db
   sqlite:///socioeconomic.db
Done.
```

Out[32]:

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

Problem 5

What kinds of crimes were recorded at schools?

In [35]:
```
%sql SELECT DISTINCT(PRIMARY_TYPE) FROM CHICAGO_CRIME_DATA WHERE LOCATION_DESCRIPTI
```

```
* sqlite:///chicago.db
  sqlite:///socioeconomic.db
Done.
```

Out[35]:

PRIMARY_TYPE
BATTERY
CRIMINAL DAMAGE
NARCOTICS
ASSAULT
CRIMINAL TRESPASS
PUBLIC PEACE VIOLATION

Problem 6

List the average safety score for each type of school.

In [37]:
```
%sql SELECT "Elementary, Middle, or High School", AVG(SAFETY_SCORE) FROM CHICAGO_PU
```

```
* sqlite:///chicago.db
  sqlite:///socioeconomic.db
Done.
```

Out[37]:

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 [39]:
```
%sql SELECT COMMUNITY_AREA_NAME, PERCENT_HOUSEHOLDS_BELOW_POVERTY FROM CENSUS_DATA
```

```
 * sqlite:///chicago.db
   sqlite:///socioeconomic.db
Done.
```

Out[39]:

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?

In [42]:
```
%%sql SELECT COMMUNITY_AREA_NUMBER , COUNT (*) AS FREQUENCY FROM CHICAGO_CRIME_DATA
GROUP BY COMMUNITY_AREA_NUMBER ORDER BY FREQUENCY DESC LIMIT 1
```

```
 * sqlite:///chicago.db
   sqlite:///socioeconomic.db
Done.
```

Out[42]:

COMMUNITY_AREA_NUMBER	FREQUENCY
25.0	43

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 [50]:
```
%%sql select community_area_name, hardship_index from census_data
where hardship_index= (select max(hardship_index) from census_data)
```

```
 * sqlite:///chicago.db
   sqlite:///socioeconomic.db
Done.
```

Out[50]:

COMMUNITY_AREA_NAME	HARDSHIP_INDEX
Riverdale	98.0

Problem 10

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

In [58]:
```
%%sql select community_area_name from census_data where community_area_number= (sel
where community_area_number= 25 limit 1)
```

```
 * sqlite:///chicago.db
   sqlite:///socioeconomic.db
Done.
```

Out[58]:

COMMUNITY_AREA_NAME
Austin

Author(s)

Hima Vasudevan

Rav Ahuja

Ramesh Sannreddy

Contribtuor(s)

Malika Singla

Change log

Date	Version	Changed by	Change Description
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