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 Db2 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: For the learners who are encountering issues with loading from .csv in DB2 on Firefox, you can download the .txt files and load the data with those:

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

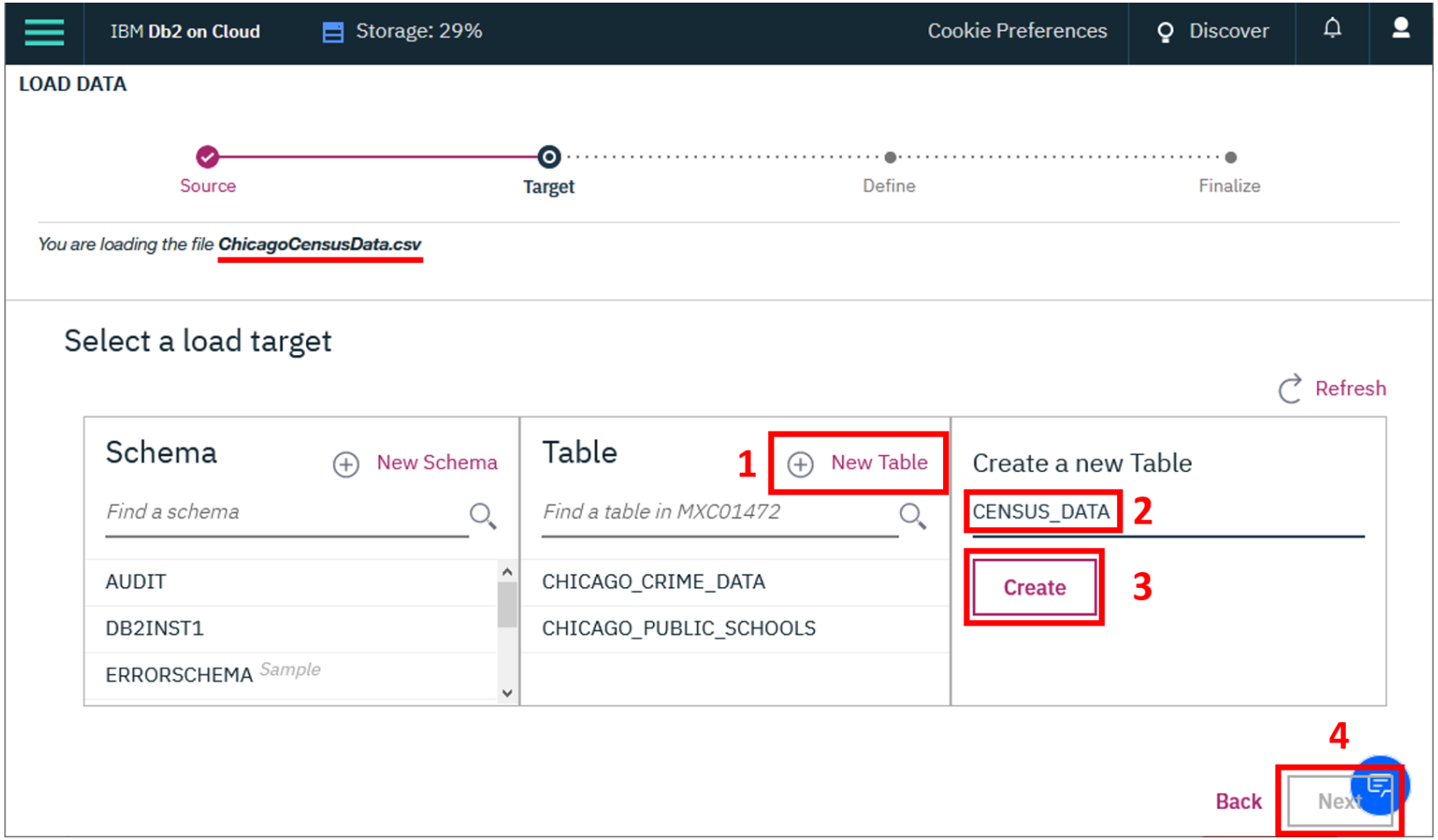
NOTE: Ensure you have downloaded the datasets using the links above instead of directly from the Chicago Data Portal. The versions linked here are subsets of the original datasets and have some of the column names modified to be more database friendly which will make it easier to complete this assignment.

Store the datasets in database tables

To analyze the data using SQL, it first needs to be stored in the database.

While it is easier to read the dataset into a Pandas dataframe and then PERSIST it into the database as we saw in Week 3 Lab 3, it results in mapping to default datatypes which may not be optimal for SQL querying. For example a long textual field may map to a CLOB instead of a VARCHAR.

Therefore, **it is highly recommended to manually load the table using the database console LOAD tool, as indicated in Week 2 Lab 1 Part II**. The only difference with that lab is that in Step 5 of the instructions you will need to click on create "(+) New Table" and specify the name of the table you want to create and then click "Next".



Now open the Db2 console, open the LOAD tool, Select / Drag the .CSV file for the first dataset, Next create a New Table, and then follow the steps on-screen instructions to load the data. Name the new tables as follows:

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

Connect to the database

Let us first load the SQL extension and establish a connection with the database

The following required modules are pre-installed in the Skills Network Labs environment. However if you run this notebook commands in a different Jupyter environment (e.g. Watson Studio or Ananconda) you may need to install these libraries by removing the `#` sign before `!pip` in the code cell below.

In [9]:
```
# These libraries are pre-installed in SN Labs. If running in another environment please uncomment lines below to ins
!pip install --force-reinstall ibm_db==3.1.0 ibm_db_sa==0.3.3
# Ensure we don't load_ext with sqlalchemy>=1.4 (incompadible)
!pip uninstall sqlalchemy==1.4 -y && pip install sqlalchemy==1.3.24
!pip install ipython-sql
```

```
Collecting ibm_db==3.1.0
  Using cached ibm_db-3.1.0-cp37-cp37m-linux_x86_64.whl
Collecting ibm_db_sa==0.3.3
  Using cached ibm_db_sa-0.3.3-py3-none-any.whl
Collecting sqlalchemy>=0.7.3
  Using cached SQLAlchemy-2.0.5.post1-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.7 MB)
Collecting importlib-metadata
  Using cached importlib_metadata-6.0.0-py3-none-any.whl (21 kB)
Collecting typing-extensions>=4.2.0
  Using cached typing_extensions-4.5.0-py3-none-any.whl (27 kB)
Collecting greenlet!=0.4.17
  Using cached greenlet-2.0.2-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (566 kB)
Collecting zipp>=0.5
  Using cached zipp-3.15.0-py3-none-any.whl (6.8 kB)
Installing collected packages: ibm_db, zipp, typing-extensions, greenlet, importlib-metadata, sqlalchemy, ibm_db_sa
  Attempting uninstall: ibm_db
    Found existing installation: ibm-db 3.1.0
    Uninstalling ibm-db-3.1.0:
      Successfully uninstalled ibm-db-3.1.0
  Attempting uninstall: zipp
    Found existing installation: zipp 3.15.0
    Uninstalling zipp-3.15.0:
      Successfully uninstalled zipp-3.15.0
  Attempting uninstall: typing-extensions
    Found existing installation: typing_extensions 4.5.0
    Uninstalling typing_extensions-4.5.0:
      Successfully uninstalled typing_extensions-4.5.0
  Attempting uninstall: greenlet
    Found existing installation: greenlet 2.0.2
    Uninstalling greenlet-2.0.2:
      Successfully uninstalled greenlet-2.0.2
  Attempting uninstall: importlib-metadata
    Found existing installation: importlib-metadata 6.0.0
    Uninstalling importlib-metadata-6.0.0:
      Successfully uninstalled importlib-metadata-6.0.0
  Attempting uninstall: sqlalchemy
    Found existing installation: SQLAlchemy 1.3.24
    Uninstalling SQLAlchemy-1.3.24:
      Successfully uninstalled SQLAlchemy-1.3.24
  Attempting uninstall: ibm_db_sa
    Found existing installation: ibm-db-sa 0.3.3
    Uninstalling ibm-db-sa-0.3.3:
```

```
      Successfully uninstalled ibm-db-sa-0.3.3
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This beha
viour is the source of the following dependency conflicts.
dash 2.7.0 requires dash-core-components==2.0.0, which is not installed.
dash 2.7.0 requires dash-html-components==2.0.0, which is not installed.
dash 2.7.0 requires dash-table==5.0.0, which is not installed.
Successfully installed greenlet-2.0.2 ibm_db-3.1.0 ibm_db_sa-0.3.3 importlib-metadata-6.0.0 sqlalchemy-2.0.5.post1 t
yping-extensions-4.5.0 zipp-3.15.0
Found existing installation: SQLAlchemy 2.0.5.post1
Uninstalling SQLAlchemy-2.0.5.post1:
  Successfully uninstalled SQLAlchemy-2.0.5.post1
Collecting sqlalchemy==1.3.24
  Using cached SQLAlchemy-1.3.24-cp37-cp37m-manylinux2010_x86_64.whl (1.3 MB)
Installing collected packages: sqlalchemy
Successfully installed sqlalchemy-1.3.24
Requirement already satisfied: ipython-sql in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (0.3.9)
Requirement already satisfied: ipython>=1.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
ipython-sql) (7.33.0)
Requirement already satisfied: ipython-genutils>=0.1.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-pack
ages (from ipython-sql) (0.2.0)
Requirement already satisfied: prettytable in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from i
python-sql) (3.5.0)
Requirement already satisfied: six in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-s
ql) (1.16.0)
Requirement already satisfied: sqlalchemy>=0.6.7 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages
(from ipython-sql) (1.3.24)
Requirement already satisfied: sqlparse in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipyt
hon-sql) (0.4.3)
Requirement already satisfied: jedi>=0.16 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ip
ython>=1.0->ipython-sql) (0.18.2)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /home/jupyterlab/conda/envs/python/li
b/python3.7/site-packages (from ipython>=1.0->ipython-sql) (3.0.33)
Requirement already satisfied: pexpect>4.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from i
python>=1.0->ipython-sql) (4.8.0)
Requirement already satisfied: pickleshare in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from i
python>=1.0->ipython-sql) (0.7.5)
Requirement already satisfied: traitlets>=4.2 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (fro
m ipython>=1.0->ipython-sql) (5.6.0)
Requirement already satisfied: backcall in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipyt
hon>=1.0->ipython-sql) (0.2.0)
Requirement already satisfied: decorator in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipy
thon>=1.0->ipython-sql) (5.1.1)
```

```
Requirement already satisfied: pygments in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipyt
hon>=1.0->ipython-sql) (2.13.0)
Requirement already satisfied: setuptools>=18.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (f
rom ipython>=1.0->ipython-sql) (65.5.1)
Requirement already satisfied: matplotlib-inline in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages
(from ipython>=1.0->ipython-sql) (0.1.6)
Requirement already satisfied: wcwidth in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from prett
ytable->ipython-sql) (0.2.5)
Requirement already satisfied: importlib-metadata in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages
(from prettytable->ipython-sql) (6.0.0)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages
(from jedi>=0.16->ipython>=1.0->ipython-sql) (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (fr
om pexpect>4.3->ipython>=1.0->ipython-sql) (0.7.0)
Requirement already satisfied: typing-extensions>=3.6.4 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-pac
kages (from importlib-metadata->prettytable->ipython-sql) (4.5.0)
Requirement already satisfied: zipp>=0.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from imp
ortlib-metadata->prettytable->ipython-sql) (3.15.0)
```

In [2]:
```
%load_ext sql
```

In the next cell enter your db2 connection string. Recall you created Service Credentials for your Db2 instance in first lab in Week 3. From your Db2 service credentials copy everything after db2:// (except the double quote at the end) and paste it in the cell below after ibm_db_sa://

```
    "db2": {
      "authentication": {
        "method": "direct",
        "password": "[redacted]",
        "username": "qdg93144"
      },
      "certificate": {
        "certificate_base64": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURFakNDQWZxZ0F3SUJBZ0lKQVA1S0R3ZTNCTkxiTUEwR0NTcUdTSWIzRFFFQkN3VUFNQjR4SERBYUJnT1YKQkFNTUUwbEN
UU0JEYkc5MVpDQkVZWFJoWW1GelpYTXdIaGNOTWpBd01qSTVNRFF5TVRBeVdoY05NekF3TWpJMgpNRFF5TVRBeVdqQWVNUnd3R2dZRFZRUUREQk5KUWswZ1EyeHZkV1FnUkdGMFlXSmhjMlZ6TUlJQklqQU5CZ2txCmhra
Uc5dzBCQVFFRkFBT0NBUThBTUlJQkNnS0NBUUVBdXUvbitpWW9xdkdGNU8xSGpEalpsK25iYjE4UkR4ZGwKTzRUL3FoUGMxMTREY1FUK0plRXdhdG13aGljTGxaQnF2QWFMb1hrbmhqSVFOMG01L0x5YzdBY291VXNmSGR
0QwpDVGcrSUsxbjBrdDMrTHM3d1dTakxqVE96N3M3M1ZUSU5yYmx3cnRIRUlvM1JWTkV6SkNHYW5LSXdZMWZVSUtrCldNMlR0SD15cnFsSGN0Z2pIUlFmRkVTRmlYaHJiODhSQmd0amIva0xtVGpCaTFBeEVadWNobWZ2Q
VRmNENOY3EKY21QcHNqdDBPTnI0YnhJMVRyUWxEemNiN1hMSFBrWW91SUprdnVzMUZvaTEySmRNM1MrK3labFZPMUZmZkU3bwpKMjhUdGJoZ3JGOGtIU0NMSkJvTTFSZ3FPZG9OVm5QOC9EOWZhamNNN01Wd2V4a0lSOTN
KR1FJREFRQUJvMU13ClVUQWRCZ05WSFE0RUZnUVVlQ3JZanFJQzc1VUpxVmZEMDh1ZWdqeDZiUmN3SHdZRFZSMFRBUUgvQkFVd0F3RUIve
kFOQmdrcWhraUc5dzBCQVFzRgpBQU9DQVFFQUkyRTBUOUt3MlN3RjJ2MXBqaHV4M0lkWWV2SGFVSkRMb0tPd0hSRnFSOHgxZ2dRcGVEcFBnMk5SCkx3R08yek85SWZUMmhLaWd1d2orWnJ5SGxxcHlxQ0pLOHJEU28xZUV
PekIyWmE2S1YrQTVscEttMWdjV3VHYzMKK1UrVTFzTDdlUjd3ZFFuVjU0TVU4aERvNi9sVHRMRVB2Mnc3VlNPSlFDK013ejgrTFJMdjVHSW5BNlJySWNhKwozM0wxNnB4ZEttd1pLYThWcnBnMXJ3QzRnY3dlYUhYMUNEW
E42K0JIbzhvWG5YWkh6UG91cldYS1BoaGdXZ2J5CkNDcUdIK0NWNnQ1eFg3b05NS3VNSUNqRVZndnNLWnRqeTQ5VW5iNVZZbHQ0b1J3dTFlbGdzRDNjekltbjlLREQKNHB1REFvYTZyMktZZE4xVkxuN3F3VG1TbDlTU05
RPT0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=",
        "name": "1cbbb1b6-3a1a-4d49-9262-3102a8f7a7c8"
      },
      "composed": [
        "db2://qdg93144:[redacted]@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb?authSource=admin&replicaSet=r
eplset"
      ],
      "database": "bludb",
      "host_tls": [
        "54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30592"
      ],
      "hosts": [
        {
          "hostname": "[redacted]",
          "port": 32733
        }
      ],
      "jdbc_url": [
        "jdbc:db2://54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb:user=<userid>;password=<your_password>;sslConnecti
on=true;"
      ],
      "path": "/bludb"
```


In [3]:
```
# Remember the connection string is of the format:
# %sql ibm_db_sa://my-username:my-password@my-hostname:my-port/my-db-name?security=SSL
# Enter the connection string for your Db2 on Cloud database instance below
%sql ibm_db_sa://nlc77149:yCbCdAmoNkScYXTP@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdom
```

Out[3]:
```
'Connected: nlc77149@bludb'
```

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 [16]:
```
%sql SELECT COUNT(CASE_NUMBER) FROM CHICAGO_CRIME_DATA;
```

```
 * ibm_db_sa://nlc77149:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

Out[16]:

1
533

Problem 2

List community areas with per capita income less than 11000.

In [19]:
```
%sql SELECT COMMUNITY_AREA_NAME, PER_CAPITA_INCOME FROM CENSUS_DATA WHERE PER_CAPITA_INCOME < 11000;
```

```
 * ibm_db_sa://nlc77149:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

Out[19]:

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 [20]:
```
%sql SELECT DISTINCT CASE_NUMBER, DESCRIPTION FROM CHICAGO_CRIME_DATA WHERE DESCRIPTION LIKE '%MINOR%';
```

```
 * ibm_db_sa://nlc77149:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

Out[20]:

case_number	description
HK238408	ILLEGAL CONSUMPTION BY MINOR
HL266884	SELL/GIVE/DEL LIQUOR TO MINOR

Problem 4

List all kidnapping crimes involving a child?

In [21]:
```
%sql SELECT DISTINCT CASE_NUMBER, DATE, DESCRIPTION FROM CHICAGO_CRIME_DATA WHERE PRIMARY_TYPE='KIDNAPPING'
```

```
 * ibm_db_sa://nlc77149:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

Out[21]:

case_number	DATE	description
HN144152	2007-01-26	CHILD ABDUCTION/STRANGER

Problem 5

What kinds of crimes were recorded at schools?

In [22]:
```
%%sql SELECT DISTINCT(PRIMARY_TYPE)
        FROM CHICAGO_CRIME_DATA
       WHERE LOCATION_DESCRIPTION LIKE '%SCHOOL%';
```

```
 * ibm_db_sa://nlc77149:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

Out[22]:

primary_type
ASSAULT
BATTERY
CRIMINAL DAMAGE
CRIMINAL TRESPA
NARCOTICS
PUBLIC PEACE VI

Problem 6

List the average safety score for each type of school.

In [26]:
```
%%sql SELECT ELEMENTARY__MIDDLE__OR_HIGH_SCHOOL type_of_school,
             AVG(SAFETY_SCORE)  as AVERAGE_SAFETY_SCORE
        FROM CHICAGO_PUBLIC_SCHOOLS
      GROUP BY  ELEMENTARY__MIDDLE__OR_HIGH_SCHOOL
```

```
 * ibm_db_sa://nlc77149:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

Out[26]:

type_of_school	average_safety_score
ES	49
HS	49
MS	48

Problem 7

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

In [27]:
```
%%sql SELECT COMMUNITY_AREA_NAME, PERCENT_HOUSEHOLDS_BELOW_POVERTY FROM CENSUS_DATA
      ORDER BY PERCENT_HOUSEHOLDS_BELOW_POVERTY DESC
      LIMIT 5
```

```
 * ibm_db_sa://nlc77149:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

Out[27]:

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 [31]:
```
%%sql SELECT community_area_number, COUNT(COMMUNITY_AREA_NUMBER) AS qtd_crimes
        FROM CHICAGO_CRIME_DATA
      GROUP BY COMMUNITY_AREA_NUMBER
      ORDER BY COUNT(COMMUNITY_AREA_NUMBER) DESC
      LIMIT 1;
```

```
 * ibm_db_sa://nlc77149:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

Out[31]:

community_area_number	qtd_crimes
25	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 [4]:
```
%%sql SELECT community_area_number, community_area_name
        FROM CENSUS_DATA
       WHERE HARDSHIP_INDEX = (SELECT max(HARDSHIP_INDEX)
                                 FROM CENSUS_DATA);
```

```
 * ibm_db_sa://nlc77149:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

Out[4]:

community_area_number	community_area_name
54	Riverdale

Problem 10

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

In [5]:
```
%%sql SELECT b.community_area_number,
             b.community_area_name,
             count(1) total_crimes
```

```
    FROM CHICAGO_CRIME_DATA a,
         CENSUS_DATA b
   WHERE a.community_area_number = b.community_area_number
  GROUP BY b.community_area_number, b.community_area_name
  ORDER BY total_crimes desc
  LIMIT 1;
```

```
 * ibm_db_sa://nlc77149:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32459/bludb
Done.
```

Out[5]:

community_area_number	community_area_name	total_crimes
25	Austin	43

Author(s)

Hima Vasudevan

Rav Ahuja

Ramesh Sannreddy

Contribtuor(s)

Malika Singla

Change log

Date	Version	Changed by	Change Description
2021-11-17	2.6	Lakshmi	Updated library
2021-05-19	2.4	Lakshmi Holla	Updated the question
2021-04-30	2.3	Malika Singla	Updated the libraries

Date	Version	Changed by	Change Description
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