



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 Anaconda) you may need to install these libraries by removing the `#` sign before `!pip` in the code cell below.

```
In [29]: # These Libraries are pre-installed in SN Labs. If running in another environment please
# !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 (incompatible)
# !pip uninstall sqlalchemy==1.4 -y && pip install sqlalchemy==1.3.24
# !pip install ipython-sql
import sqlite3
```

```
con = sqlite3.connect("RealWorldData.db")
cur = con.cursor()
```

In [30]: !pip install -q pandas==1.1.5

In [5]: %load_ext sql

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

In [31]: 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="multi")

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/gener
c.py:2882: UserWarning: The spaces in these column names will not be changed. In p
andas versions < 0.14, spaces were converted to underscores.
both result in 0.1234 being formatted as 0.12.

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": "",
        "username": "qdg93144"
    },
    "certificate": [
        "certificate_base64": "LS0tLS1CRUdJTiBDRFJUSUZjQ0FURS0tLs9tCk1JSURFaikNDQwZxZ0F3SUJB201KQVA1S0R3ZTNTCKxiTUEwR0NtCeUdTSwIzRFFFQkN3VUFNQjR4SERBYUjnT1YKQkFNTUUvbEN  

UU03EYkc5WfJchNW1GelpYTdxiaGNTWpb901c5tVNRFF5TRBRevdoy05Nek3T1kpJMsgrNRF5TWR8eVcqjWuNund3P2gZFRZRUREQk5KUkewzI1jye0ZkV1FnUkeGMF1XSmhjM1Z6TU1j0k1nQUSC2ZtXChhx  

Uc5cZbCOVFRkBTQb1NtJQKwNSBNBUWbxUvibp9wxdGySGPea1ps253YjE4UkRAZGwTzRUL3Fj0lGKmTREY1FK0p1RXh0G13a1G1jTxGx0nF2QWFMD1hbmnhsVFOMG01L0s5YzcbV91XWnSGR  

0qwpDVGcsu1sxsBzrdMzTH3idTaKxv9Ew94N3M1ZUSUy9mx3cR1RuluM1jWTKw65kHyW1LSXz2MwZvSUt2r1dNM1r8S15cFSGN022pIU1FnRkVTrn1yah1j00hS0m0a0iav6x+VgpcCaTFB8eVadivlobiZQ2  

VRmDNOY3EKY21chNqo0BPTnT0YnhJMWryUwxEamNi1MSFBxw91S1pxdnVzMuVaTeY5mRN1MyK31abZPMUz2Mku3bpwKjhudJo23Z3G0tIU0NMSk1vTFFS23FPZG90Vm5009EWZhamaNNN01wd2V4a01S0TN  

KR1fJREFR0JUJMi13C1VU0W4P5WSFEBRUzluV1032JanF0jzc1V1UpxVnZEMD12WdqoDz1uN3SHd2ZRFZSMGpc0md3Rg9BVvVc1kkfanF3j0c1VpxvZEMD1h2WdqoD2ziUm3Rhd2ZFSMF8BUUgvQkFv0f3R1Uve  

KF0dmrcWhxaLcGz2BCm0VekzrgoBQ9UQVFF0ukyRTBu0U31N3Rj3jMXBqaV4M0ikwV25GFVSKRmbd0hSrnfSnGz22dCrGeVcBnMgSSCx3R08yeK85SwUZMmlLwA012orwJ3SSGxxch1x0qL0HjeU28xZUV  

PeKlymE2S1YjQVs=e7tMdj3VHYzMKKUuVTFtTB01uj3Z2FFUyU0FTVU4aRvN19sVHMRVB2Mc3SVNPS1FD013ejgjzTFJMdVHSw5Bn1jY5whkwozMoWxNn4Zett01pLYThwcnBrMxJ3Q2rnY31vUhYMUNEW  

E42K01IbzhiW6h6UG91clad0aGxZ2J5ckDcUoIK0NNWQ1leFgb05N3VNUNqRVZndnNLWnRqeT05Vn5iNVZzbhQ0b1j3dT1bgdzRDNje1kb1j1LREQKHN81REFyV7zyMktZZE4xVkuN3F3VG1TbD1UT05  

RPTOKL50tS1FTQgQ0VSVEIGSUNBVUtlS0tLQo=",
    "name": "icbb1b6-3ala-d4d-9262-3102a8f7a7cW"
},
"composed": [
    "db2://qdg93144:54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd@tgtu01qde00.databases.appdomain.cloud:32733/bludb?authSource=admin&replicaSet=r
epset"
],
"database": "bludb",
"host_ip": "54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd@tgtu01qde00.databases.appdomain.cloud:30592",
"hosts": [
    {
        "hostname": "",
        "port": 32733
    }
],
"jdbc_url": [
    "jdbc:db2://54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd@tgtu01qde00.databases.appdomain.cloud:32733/bludb:userid=<userid>;password=<your_password>;sslConnecti
on=true"
]
```

In [52]: # Enter the connection string for your Db2 on Cloud database instance below
%sql ibm_db_sa://my-username:my-password@my-hostname:my-port/my-db-name?security=
%sql ibm_db_sa://hyh87123:QkGTIBFzSctAHbdQ@764264db-9824-4b7c-82df-40d1b13897c2.bs2

Out[52]: 'Connected: hyh87123@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 [71]: %sql SELECT COUNT(*) AS TOTAL_CRIMES \
    FROM CHICAGO_CRIME_DATA
* ibm_db_sa://hyh87123:**@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8
lcg.databases.appdomain.cloud:32536/bludb
Done.

Out[71]: total_crimes
533
```

Problem 2

List community areas with per capita income less than 11000.

```
In [34]: %sql SELECT COMMUNITY_AREA_NAME FROM CENSUS_DATA WHERE PER_CAPITA_INCOME < 11000;
* ibm_db_sa://hyh87123:**@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8
lcg.databases.appdomain.cloud:32536/bludb
Done.

Out[34]: 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 [54]: %%sql
/*Problem 3: List all case numbers for crimes involving minors?*/
SELECT DISTINCT CASE_NUMBER from CHICAGO_CRIME_DATA
where DESCRIPTION like '%MINOR%'

* ibm_db_sa://hyh87123:**@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8
lcg.databases.appdomain.cloud:32536/bludb
Done.

Out[54]: case_number
HK238408
HL266884
```

Problem 4

List all kidnapping crimes involving a child?

```
In [55]: %sql SELECT DISTINCT CASE_NUMBER, PRIMARY_TYPE, DATE, DESCRIPTION FROM CHICAGO_CRIM
WHERE PRIMARY_TYPE = 'KIDNAPPING'
```

```
* ibm_db_sa://hyh87123:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8
lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

Problem 5

What kinds of crimes were recorded at schools?

```
In [60]: %sql SELECT DISTINCT(PRIMARY_TYPE), LOCATION_DESCRIPTION FROM CHICAGO_CRIME_DATA \
WHERE LOCATION_DESCRIPTION LIKE '%SCHOOL%'
```

```
* ibm_db_sa://hyh87123:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8
lcg.databases.appdomain.cloud:32536/bludb
Done.
```

primary_type	location_description
PUBLIC PEACE VI	SCHOOL, PRIVATE, BUILDING
BATTERY	SCHOOL, PUBLIC, BUILDING
NARCOTICS	SCHOOL, PUBLIC, BUILDING
PUBLIC PEACE VI	SCHOOL, PUBLIC, BUILDING
ASSAULT	SCHOOL, PUBLIC, GROUNDS
BATTERY	SCHOOL, PUBLIC, GROUNDS
CRIMINAL DAMAGE	SCHOOL, PUBLIC, GROUNDS
CRIMINAL TRESPA	SCHOOL, PUBLIC, GROUNDS
NARCOTICS	SCHOOL, PUBLIC, GROUNDS

Problem 6

List the average safety score for each type of school.

```
In [69]: %sql SELECT "Elementary, Middle, or High School", AVG(安全感分) AVERAGE_SAFETY_
FROM CHICAGO_PUBLIC_SCHOOLS GROUP BY "Elementary, Middle, or High School" ;
```

```
* ibm_db_sa://hyh87123:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8
lcg.databases.appdomain.cloud:32536/bludb
(ibm_db_dbi.ProgrammingError) ibm_db_dbi::ProgrammingError: SQLNumResultCols failed
d: [IBM][CLI Driver][DB2/LINUXX8664] SQL0206N "Elementary, Middle, or High School"
1" is not valid in the context where it is used. SQLSTATE=42703 SQLCODE=-206
[SQL: SELECT "Elementary, Middle, or High School", AVG(SAFETY_SCORE) AVERAGE_SAFETY_SCORE
FROM CHICAGO_PUBLIC_SCHOOLS GROUP BY "Elementary, Middle, or High School"
;]
(Background on this error at: http://sqlalche.me/e/13/f405)
```

Problem 7

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

In [62]: `%%sql`

```
SELECT COMMUNITY_AREA_NAME, PERCENT_HOUSEHOLDS_BELOW_POVERTY
FROM CENSUS_DATA
ORDER BY PERCENT_HOUSEHOLDS_BELOW_POVERTY DESC
LIMIT 5 ;
```

```
* ibm_db_sa://hyh87123:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8
lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[62]: `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 [63]: `%%sql`

```
SELECT CCD.COMMUNITY_AREA_NUMBER ,COUNT(CCD.COMMUNITY_AREA_NUMBER) AS FREQUENCY
FROM CHICAGO_CRIME_DATA AS CCD
GROUP BY CCD.COMMUNITY_AREA_NUMBER
ORDER BY COUNT(CCD.COMMUNITY_AREA_NUMBER) DESC
LIMIT 1;
```

```
* ibm_db_sa://hyh87123:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8
lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[63]: `community_area_number frequency`

community_area_number	frequency
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 [64]: `%%sql`

```
SELECT COMMUNITY_AREA_NAME
FROM CENSUS_DATA
WHERE HARSHSHIP_INDEX = (SELECT MAX(HARSHSHIP_INDEX) FROM CENSUS_DATA);
```

* ibm_db_sa://hyh87123:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8
lcg.databases.appdomain.cloud:32536/bludb
Done.

Out[64]: `community_area_name`

community_area_name
Riverdale

Problem 10

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

In [65]: `%%sql`

```
SELECT community_area_name
FROM CENSUS_DATA
WHERE COMMUNITY_AREA_NUMBER = (
    SELECT CCD.COMMUNITY_AREA_NUMBER
    FROM CHICAGO_CRIME_DATA AS CCD
    GROUP BY CCD.COMMUNITY_AREA_NUMBER
    ORDER BY COUNT(CCD.COMMUNITY_AREA_NUMBER) DESC
    LIMIT 1)

LIMIT 1;
```

* ibm_db_sa://hyh87123:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8
lcg.databases.appdomain.cloud:32536/bludb
Done.

Out[65]: `community_area_name`

community_area_name
Austin

Copyright © 2020 cognitiveclass.ai. This notebook and its source code are released under the terms of the [MIT License](#).

Author(s)

Hima Vasudevan

Rav Ahuja

Ramesh Sannreddy

Contributor(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
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 2020. All rights reserved.