DB0201EN-PeerAssign-v5

March 15, 2023

Assignment: Notebook for Peer Assignment

1 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

1.1 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

1.1.1 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

1.1.2 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

1.1.3 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

1.1.4 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.

1.1.5 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**

1.1.6 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.

[ ]:
```
# These libraries are pre-installed in SN Labs. If running in another␣
 ↪environment please uncomment lines below to install them:
# !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
```

[24]:
```
%load_ext sql
```

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

[25]:
```
# 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#␣
 ↪1754175317511752
%sql ibm_db_sa://pvm06411:2qJwl8PKKPXhLAsw@764264db-9824-4b7c-82df-40d1b13897c2.
 ↪bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb?security=SSL
```

[25]:
```
'Connected: pvm06411@bludb'
```

1.2 Problems

Now write and execute SQL queries to solve assignment problems

1.2.1 Problem 1

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

[26]:
```
%sql select count(distinct ID) as Count from CHICAGO_CRIME_DATA
```

```
 * ibm_db_sa://pvm06411:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

[26]:
```
[(533,)]
```

1.2.2 Problem 2

List community areas with per capita income less than 11000.

[ ]:
```
%sql SELECT community_area_name, per_capita_income FROM CENSUS_DATA WHERE␣
↪PER_CAPITA_INCOME<11000
```

```
 * ibm_db_sa://pvm06411:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

[ ]:
```
[('West Garfield Park', 10934),
 ('South Lawndale', 10402),
 ('Fuller Park', 10432),
 ('Riverdale', 8201)]
```

1.2.3 Problem 3

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

[67]:
```
%sql select case_number from CHICAGO_CRIME_DATA where DESCRIPTION like '%MINOR%'
```

```
 * ibm_db_sa://pvm06411:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

[67]:
```
[('HL266884',), ('HK238408',)]
```

1.2.4 Problem 4

List all kidnapping crimes involving a child?

[36]:
```
%sql select case_number from CHICAGO_CRIME_DATA where primary_type='KIDNAPPING'␣
↪and description like '%CHILD%';
```

```
 * ibm_db_sa://pvm06411:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

[36]:
```
[('HN144152',)]
```

1.2.5 Problem 5

What kinds of crimes were recorded at schools?

[65]:
```
%sql select distinct primary_type from CHICAGO_CRIME_DATA where␣
↪LOCATION_DESCRIPTION like '%SCHOOL%';
```

```
 * ibm_db_sa://pvm06411:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

[65]:
```
[('ASSAULT',),
 ('BATTERY',),
 ('CRIMINAL DAMAGE',),
 ('CRIMINAL TRESPA',),
 ('NARCOTICS',),
 ('PUBLIC PEACE VI',)]
```

1.2.6 Problem 6

List the average safety score for each type of school.

[64]:
```
%sql SELECT ELEMENTARY__MIDDLE__OR_HIGH_SCHOOL, AVG(Cast(SAFETY_SCORE as␣
↪Float)) AS AVERAGE_SAFETY_SCORE FROM CHICAGO_PUBLIC_SCHOOLS WHERE␣
↪ELEMENTARY__MIDDLE__OR_HIGH_SCHOOL IN ('ES', 'MS', 'HS') GROUP BY␣
↪ELEMENTARY__MIDDLE__OR_HIGH_SCHOOL;
```

```
 * ibm_db_sa://pvm06411:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

[64]:
```
[('ES', 49.52038369304557), ('HS', 49.62352941176471), ('MS', 48.0)]
```

1.2.7 Problem 7

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

[52]:
```
%sql SELECT COMMUNITY_AREA_NAME, PERCENT_HOUSEHOLDS_BELOW_POVERTY FROM␣
↪CENSUS_DATA ORDER BY PERCENT_HOUSEHOLDS_BELOW_POVERTY DESC LIMIT 5;
```

```
 * ibm_db_sa://pvm06411:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

[52]:
```
[('Riverdale', Decimal('56.5')),
 ('Fuller Park', Decimal('51.2')),
 ('Englewood', Decimal('46.6')),
 ('North Lawndale', Decimal('43.1')),
 ('East Garfield Park', Decimal('42.4'))]
```

1.2.8 Problem 8

Which community area is most crime prone?

[60]:
```
%sql SELECT COMMUNITY_AREA_NUMBER, COUNT(*) as CRIME_COUNT FROM␣
↪CHICAGO_CRIME_DATA GROUP BY COMMUNITY_AREA_NUMBER ORDER BY CRIME_COUNT DESC␣
↪LIMIT 1;
```

```
 * ibm_db_sa://pvm06411:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

[60]:
```
[(25, 43)]
```

Double-click **here** for a hint

1.2.9 Problem 9

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

[62]:
```
%sql SELECT COMMUNITY_AREA_NAME FROM CENSUS_DATA WHERE HARDSHIP_INDEX = (SELECT␣
↪MAX(HARDSHIP_INDEX) FROM CENSUS_DATA);
```

```
 * ibm_db_sa://pvm06411:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

[62]:
```
[('Riverdale',)]
```

1.2.10 Problem 10

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

[63]:
```
%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);
```

```
 * ibm_db_sa://pvm06411:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

[63]:
```
[('Austin',)]
```


1.3 Author(s)

Hima Vasudevan

Rav Ahuja

Ramesh Sannreddy

1.4 Contribtuor(s)

Malika Singla

1.5 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

##