Working with real Datasets using Python and SQL

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

Table of Content

I . Understand Chicago Datasets. II . Load the 3 datasets into 3 tables in a Db2 database. III . Execute SQL queries to ansqer assignmet questions.

Download the 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):

CENSUS_DATA: https://ibm.box.com/shared/static/05c3415cbfbtfnr2fx4atenb2sd361ze.csv (https://ibm.box.com/shared/static/05c3415cbfbtfnr2fx4atenb2sd361ze.csv) CHICAGO_PUBLIC_SCHOOLS https://ibm.box.com/shared/static/f9gjvj1gjmxxzycdhplzt01qtz0s7ew7.csv (https://ibm.box.com/shared/static/f9gjvj1gjmxxzycdhplzt01qtz0s7ew7.csv) CHICAGO_CRIME_DATA: https://ibm.box.com/shared/static/svflyugsr9zbqy5bmowgswqemfpm1x7f.csv (https://ibm.box.com/shared/static/svflyugsr9zbqy5bmowgswqemfpm1x7f.csv) 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.

Connect to the database

In [2]:
```
!pip install ipython-sql
```

```
Requirement already satisfied: ipython-sql in c:\users\edson\anaconda3\lib\si
te-packages (0.4.1)
Requirement already satisfied: six in c:\users\edson\anaconda3\lib\site-packa
ges (from ipython-sql) (1.16.0)
Requirement already satisfied: ipython-genutils>=0.1.0 in c:\users\edson\anac
onda3\lib\site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: ipython>=1.0 in c:\users\edson\anaconda3\lib\s
ite-packages (from ipython-sql) (8.2.0)
Requirement already satisfied: sqlalchemy>=0.6.7 in c:\users\edson\anaconda3\
lib\site-packages (from ipython-sql) (1.3.24)
Requirement already satisfied: prettytable<1 in c:\users\edson\anaconda3\lib\
```

In [3]:
```
!pip uninstall sqlalchemy==1.4.x -y && pip install sqlalchemy==1.3.24
```

```
Found existing installation: SQLAlchemy 1.3.24
Uninstalling SQLAlchemy-1.3.24:
  Successfully uninstalled SQLAlchemy-1.3.24
Collecting sqlalchemy==1.3.24
  Using cached SQLAlchemy-1.3.24-cp39-cp39-win_amd64.whl (1.2 MB)
Installing collected packages: sqlalchemy
Successfully installed sqlalchemy-1.3.24
```

In [58]:
```
!pip install ibm_db
!pip install ibm_db_sa
```

```
Requirement already satisfied: ibm_db in c:\users\edson\anaconda3\lib\site-pa
ckages (3.1.0)
Requirement already satisfied: ibm_db_sa in c:\users\edson\anaconda3\lib\site
-packages (0.3.3)
Requirement already satisfied: sqlalchemy>=0.7.3 in c:\users\edson\anaconda3\
lib\site-packages (from ibm_db_sa) (1.3.24)
```

In [4]:
```
%load_ext sql
```

In [5]:
```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import ibm_db
```

Connecting with database

In [6]:
```
dsn_hostname = "ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.data
dsn_uid = "dmq73837"        # e.g. "abc12345"
dsn_pwd = "A6ZZPRoaOUUNvI1J"      # e.g. "7dBZ3wWt9XN6$o0J"

dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "bludb"            # e.g. "BLUDB"
dsn_port = "31321"                # e.g. "32733"
dsn_protocol = "TCPIP"            # i.e. "TCPIP"
dsn_security = "SSL"              #i.e. "SSL"
```

In [7]:
```
dsn = (
    "DRIVER={0};"
    "DATABASE={1};"
    "HOSTNAME={2};"
    "PORT={3};"
```

```
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname, dsn_port,

#print the connection string to check correct values are specified
print(dsn)
```

```
DRIVER={IBM DB2 ODBC DRIVER};DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-8fc0-
d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=31321;PROTOC
OL=TCPIP;UID=dmq73837;PWD=A6ZZPRoaOUUNvI1J;SECURITY=SSL;
```

In [8]:
```
try:
    conn = ibm_db.connect(dsn, "", "")
    print ("Connected to database: ", dsn_database, "as user: ", dsn_uid, "on

except:
    print ("Unable to connect: ", ibm_db.conn_errormsg() )
```

```
Connected to database:  bludb as user:  dmq73837 on host:  ba99a9e6-d59e-4883
-8fc0-d6a8c9f7a08f.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud
```

In [9]:
```
server = ibm_db.server_info(conn)
```

In [10]:
```
print ("DBMS_NAME: ", server.DBMS_NAME)
print ("DBMS_VER:  ", server.DBMS_VER)
print ("DB_NAME:   ", server.DB_NAME)
```

```
DBMS_NAME:  DB2/LINUXX8664
DBMS_VER:   11.05.0700
DB_NAME:    BLUDB
```

In [11]:
```
%sql ibm_db_sa://dmq73837:A6ZZPRoaOUUNvI1J@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08
```

Knowing the databases

In [12]:
```
#you can retrieve list of all tables where the schema name is not one of the s

%sql select TABSCHEMA, TABNAME, CREATE_TIME from SYSCAT.TABLES \
    where TABSCHEMA not in ('SYSIBM', 'SYSCAT', 'SYSSTAT', 'SYSIBMADM', 'SYS
```

```
 * ibm_db_sa://dmq73837:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgt
u0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

Out[12]:

tabschema	tabname	create_time
DMQ73837	COUNTRY	2022-07-19 01:31:23.725660
DMQ73837	INSTRUTOR	2022-07-19 02:22:56.550329
DMQ73837	JOB_HISTORY	2022-07-20 02:25:09.478340
DMQ73837	JOBS	2022-07-20 02:25:09.706620
DMQ73837	DEPARTMENTS	2022-07-20 02:25:09.916382
DMQ73837	LOCATIONS	2022-07-20 02:25:10.160229
DMQ73837	PETSALE	2022-07-21 03:07:54.944882

DMQ73837	EMPLOYEES	2022-07-21 01:51:55.889979
DMQ73837	INSTRUCTOR	2022-07-22 01:51:42.080644
DMQ73837	INTERNATIONAL_STUDENT_TEST_SCORES	2022-07-23 23:13:20.901051
DMQ73837	MERCADO	2022-07-24 02:04:37.054903
DMQ73837	CHICAGO_CRIME	2022-07-25 03:34:33.873604
DMQ73837	CHICAGO_DATA	2022-07-24 18:42:44.683593
DMQ73837	CHICAGO_SCHOOL	2022-07-26 23:14:02.592299

In [13]:
```
%sql select distinct(NAME), COLTYPE, LENGTH from SYSIBM.SYSCOLUMNS where TBNAM
```

```
 * ibm_db_sa://dmq73837:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgt
u0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

Out[13]:

name	coltype	length
ADEQUATE_YEARLY_PROGRESS_MADE_____	VARCHAR	16
CITY	VARCHAR	8
COLLABORATIVE_NAME	VARCHAR	47
COLUMN_12	VARCHAR	7
COLUMN_13	VARCHAR	9
COLUMN_14	VARCHAR	7
COLUMN_15	VARCHAR	15
LINK	VARCHAR	80
NAME_SCHOOL	VARCHAR	64
NETWORK_MANAGER	VARCHAR	40
PHONE_NUMBER	VARCHAR	19
SCHOOL_ID	VARCHAR	13
STATE	VARCHAR	8
STREET_ADDRESS	VARCHAR	26
TYPE	VARCHAR	8
ZIP_CODE	VARCHAR	12

PROBLEMS

1 - Find the total number of crimes recorded in the CRIME table

In [14]:
```
%%sql
SELECT COUNT(*) AS NUMBER_CRIMES  FROM CHICAGO_CRIME
```

```
 * ibm_db_sa://dmq73837:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgt
u0lqde00.databases.appdomain.cloud:31321/BLUDB
```

Out[14]:

number_crimes
533

2 - Retrieve first 10 rows from the CRIME table

In [16]:
```
%sql SELECT * FROM CHICAGO_CRIME_FETCH first 10 rows ONLY;
```

```
 * ibm_db_sa://dmq73837:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgt
u0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

Out[16]:

id	case_number	DATE	block	iucr	primary_type	description	locatio
3512276	HK587712	2004-08-28	047XX S KEDZIE AVE	890	THEFT	FROM BUILDING	SMALL R
3406613	HK456306	2004-06-26	009XX N CENTRAL PARK AVE	820	THEFT	$500 AND UNDER	
8002131	HT233595	2011-04-04	043XX S WABASH AVE	820	THEFT	$500 AND UNDER	HOME/
7903289	HT133522	2010-12-30	083XX S KINGSTON AVE	840	THEFT	FINANCIAL ID THEFT: OVER $300	
10402076	HZ138551	2016-02-02	033XX W 66TH ST	820	THEFT	$500 AND UNDER	
7732712	HS540106	2010-09-29	006XX W CHICAGO AVE	810	THEFT	OVER $500	LOT/GARAGE
10769475	HZ534771	2016-11-30	050XX N KEDZIE AVE	810	THEFT	OVER $500	
4494340	HL793243	2005-12-16	005XX E PERSHING RD	860	THEFT	RETAIL THEFT	GROCERY
3778925	HL149610	2005-01-28	100XX S WASHTENAW AVE	810	THEFT	OVER $500	
3324217	HK361551	2004-05-13	033XX W BELMONT AVE	820	THEFT	$500 AND UNDER	SMALL R

3 - How many crimes involve an arrest?

In [17]:
```
%sql SELECT COUNT (ARREST) FROM CHICAGO_CRIME WHERE ARREST=TRUE;
```

```
 * ibm_db_sa://dmq73837:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgt
```

Out[17]:

1
163

4 - Which unique types of crimes have been recorded at GAS STATION locations?

In [118]:
```
%sql SELECT DISTINCT(PRIMARY_TYPE) FROM CHICAGO_CRIME WHERE LOCATION_DESCRIPTI
```

```
 * ibm_db_sa://dmq73837:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgt
u0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

Out[118]:

primary_type
CRIMINAL TRESPA
NARCOTICS
ROBBERY
THEFT

5 - In the CENUS_DATA table list all Community Areas whose names start with the letter ‘B’.

In [18]:
```
%sql SELECT COMMUNITY_AREA_NAME FROM CHICAGO_DATA WHERE COMMUNITY_AREA_NAME LI
```

```
 * ibm_db_sa://dmq73837:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgt
u0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

Out[18]:

community_area_name
Belmont Cragin
Burnside
Brighton Park
Bridgeport
Beverly

6 - Show the table Checago School.

In [20]:
```
%sql select * from CHICAGO_SCHOOL;
```

```
 * ibm_db_sa://dmq73837:***@ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.c1ogj3sd0tgt
u0lqde00.databases.appdomain.cloud:31321/BLUDB
Done.
```

Out[20]:

school_id	name_school	TYPE	street_address	city	state	zip_code	phone_number	
610038	Abraham Lincoln Elementary School	ES	615 W Kemper Pl	Chicago	IL	60614	(773) 534-5720	http /Sch /Spr

	610281	Adam Clayton Powell Paideia Community Academy Elementary School	ES	7511 S South Shore Dr	Chicago	IL	60649	(773) 535-6650	http /Sch /Spr
	610185	Adlai E Stevenson Elementary	ES	8010 S Kostner Ave	Chicago	IL	60652	(773) 535-2280	http /Sch /Spr

In [ ]: