DB0201EN-Week4-1-1-RealDataPractice-v4-py

October 26, 2019

Lab: Working with a real world data-set using SQL and Python

1 Introduction

This notebook shows how to work with a real world dataset using SQL and Python. In this lab you will: 1. Understand the dataset for Chicago Public School level performance 1. Store the dataset in an Db2 database on IBM Cloud instance 1. Retrieve metadata about tables and columns and query data from mixed case columns 1. Solve example problems to practice your SQL skills including using built-in database functions

1.1 Chicago Public Schools - Progress Report Cards (2011-2012)

The city of Chicago released a dataset showing all school level performance data used to create School Report Cards for the 2011-2012 school year. The dataset is available from the Chicago Data Portal: https://data.cityofchicago.org/Education/Chicago-Public-Schools-Progress-Report-Cards-2011-/9xs2-f89t

This dataset includes a large number of metrics. Start by familiarizing yourself with the types of metrics in the database: https://data.cityofchicago.org/api/assets/AAD41A13-BE8A-4E67-B1F5-86E711E09D5F?download=true

NOTE: Do not download the dataset directly from City of Chicago portal. Instead download a more database friendly version from the link below. Now download a static copy of this database and review some of its contents: https://ibm.box.com/shared/static/f9gjvj1gjmxxzycdhplzt01qtz0s7ew7.csv

1.1.1 Store the dataset in a Table

In many cases the dataset to be analyzed is available as a .CSV (comma separated values) file, perhaps on the internet. 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 the previous lab, 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 CHICAGO PUBLIC SCHOOLS dataset and load the dataset into a new table called SCHOOLS.

1.1.2 Connect to the database

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

[8]:
```
%load_ext sql
```

[9]:
```
# 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
%sql ibm_db_sa://mxl17625:ngp3l2w%2Bgkg206d9@dashdb-txn-sbox-yp-lon02-02.
 ↪services.eu-gb.bluemix.net:50000/BLUDB
```

[9]:
```
'Connected: mxl17625@BLUDB'
```

1.1.3 Query the database system catalog to retrieve table metadata

You can verify that the table creation was successful by retrieving the list of all tables in your schema and checking whether the SCHOOLS table was created

[15]:
```
# type in your query to retrieve list of all tables in the database for your␣
 ↪db2 schema (username)

%sql select TABSCHEMA, TABNAME, CREATE_TIME from syscat.tables where tabschema␣
 ↪= 'mxl17625';
```

```
 * ibm_db_sa://mxl17625:***@dashdb-txn-sbox-yp-lon02-02.services.eu-
gb.bluemix.net:50000/BLUDB
Done.
```

[15]:
```
[]
```

Double-click **here** for a hint

Double-click **here** for the solution.

1.1.4 Query the database system catalog to retrieve column metadata

The SCHOOLS table contains a large number of columns. How many columns does this table have?

[19]:
```
# type in your query to retrieve the number of columns in the SCHOOLS table

%sql select count(*) from syscat.columns where tabname = 'schools';

# %sql select distinct(name), coltype, length from sysibm.syscolumns where␣
 ↪tabname = 'schools';
```

```
 * ibm_db_sa://mxl17625:***@dashdb-txn-sbox-yp-lon02-02.services.eu-
gb.bluemix.net:50000/BLUDB
Done.
```

[19]:
```
[(Decimal('0'),)]
```

Double-click **here** for a hint

Double-click **here** for the solution.

Now retrieve the the list of columns in SCHOOLS table and their column type (datatype) and length.

[ ]:
```
# type in your query to retrieve all column names in the SCHOOLS table along␣
↪with their datatypes and length
```

Double-click **here** for the solution.

1.1.5 Questions

1. Is the column name for the "SCHOOL ID" attribute in upper or mixed case?
2. What is the name of "Community Area Name" column in your table? Does it have spaces?
3. Are there any columns in whose names the spaces and paranthesis (round brackets) have been replaced by the underscore character "_"?

1.2 Problems

1.2.1 Problem 1

How many Elementary Schools are in the dataset?

[ ]:
```

```

Double-click **here** for a hint

Double-click **here** for another hint

Double-click **here** for the solution.

1.2.2 Problem 2

What is the highest Safety Score?

[ ]:
```

```

Double-click **here** for a hint

Double-click **here** for the solution.

1.2.3 Problem 3

Which schools have highest Safety Score?

[ ]:
```

```

Double-click **here** for the solution.

1.2.4 Problem 4

What are the top 10 schools with the highest "Average Student Attendance"?

[28]:
```
%sql select Name_of_School, Average_Student_Attendance from SCHOOLS \
    order by Average_Student_Attendance desc nulls last limit 10
```

```
 * ibm_db_sa://mxl17625:***@dashdb-txn-sbox-yp-lon02-02.services.eu-
gb.bluemix.net:50000/BLUDB
Done.
```

[28]:
```
[('John Charles Haines Elementary School', '98.40%'),
 ('James Ward Elementary School', '97.80%'),
 ('Edgar Allan Poe Elementary Classical School', '97.60%'),
 ('Orozco Fine Arts & Sciences Elementary School', '97.60%'),
 ('Rachel Carson Elementary School', '97.60%'),
 ('Annie Keller Elementary Gifted Magnet School', '97.50%'),
 ('Andrew Jackson Elementary Language Academy', '97.40%'),
 ('Lenart Elementary Regional Gifted Center', '97.40%'),
 ('Disney II Magnet School', '97.30%'),
 ('John H Vanderpoel Elementary Magnet School', '97.20%')]
```

Double-click **here** for the solution.

1.2.5 Problem 5

Retrieve the list of 5 Schools with the lowest Average Student Attendance sorted in ascending order based on attendance

[32]:
```
%sql SELECT Name_of_School, Average_Student_Attendance  \
     from SCHOOLS \
     order by Average_Student_Attendance nulls last limit 5
```

```
 * ibm_db_sa://mxl17625:***@dashdb-txn-sbox-yp-lon02-02.services.eu-
gb.bluemix.net:50000/BLUDB
Done.
```

[32]:
```
[('Richard T Crane Technical Preparatory High School', '57.90%'),
 ('Barbara Vick Early Childhood & Family Center', '60.90%'),
 ('Dyett High School', '62.50%'),
 ('Wendell Phillips Academy High School', '63.00%'),
 ('Orr Academy High School', '66.30%')]
```

Double-click **here** for the solution.

1.2.6 Problem 6

Now remove the '%' sign from the above result set for Average Student Attendance column

[10]:
```
%sql SELECT Name_of_School, REPLACE(Average_Student_Attendance, '%', '') \
     from SCHOOLS \
     order by Average_Student_Attendance \
     fetch first 5 rows only
```

```
 * ibm_db_sa://mxl17625:***@dashdb-txn-sbox-yp-lon02-02.services.eu-
gb.bluemix.net:50000/BLUDB
Done.
```

[10]:
```
[('Richard T Crane Technical Preparatory High School', '57.90'),
 ('Barbara Vick Early Childhood & Family Center', '60.90'),
 ('Dyett High School', '62.50'),
 ('Wendell Phillips Academy High School', '63.00'),
 ('Orr Academy High School', '66.30')]
```

Double-click **here** for a hint

Double-click **here** for the solution.

1.2.7 Problem 7

Which Schools have Average Student Attendance lower than 70%?

[16]:
```
%sql SELECT Name_of_School, Average_Student_Attendance  \
     from SCHOOLS \
     where decimal(replace(Average_Student_Attendance,'%','')) < 70 \
     order by Average_Student_Attendance
```

```
 * ibm_db_sa://mxl17625:***@dashdb-txn-sbox-yp-lon02-02.services.eu-
gb.bluemix.net:50000/BLUDB
Done.
```

[16]:
```
[('Richard T Crane Technical Preparatory High School', '57.90%'),
 ('Barbara Vick Early Childhood & Family Center', '60.90%'),
 ('Dyett High School', '62.50%'),
 ('Wendell Phillips Academy High School', '63.00%'),
 ('Orr Academy High School', '66.30%'),
 ('Manley Career Academy High School', '66.80%'),
 ('Chicago Vocational Career Academy High School', '68.80%'),
 ('Roberto Clemente Community Academy High School', '69.60%')]
```

Double-click **here** for a hint

Double-click **here** for another hint

Double-click **here** for the solution.

1.2.8 Problem 8

Get the total College Enrollment for each Community Area

[17]:
```
%sql select Community_Area_Name, sum(College_Enrollment) AS TOTAL_ENROLLMENT \
   from SCHOOLS \
   group by Community_Area_Name
```

```
 * ibm_db_sa://mxl17625:***@dashdb-txn-sbox-yp-lon02-02.services.eu-
gb.bluemix.net:50000/BLUDB
Done.
```

[17]:
```
[('ALBANY PARK', 6864),
 ('ARCHER HEIGHTS', 4823),
 ('ARMOUR SQUARE', 1458),
 ('ASHBURN', 6483),
 ('AUBURN GRESHAM', 4175),
 ('AUSTIN', 10933),
 ('AVALON PARK', 1522),
 ('AVONDALE', 3640),
 ('BELMONT CRAGIN', 14386),
 ('BEVERLY', 1636),
 ('BRIDGEPORT', 3167),
 ('BRIGHTON PARK', 9647),
 ('BURNSIDE', 549),
 ('CALUMET HEIGHTS', 1568),
 ('CHATHAM', 5042),
 ('CHICAGO LAWN', 7086),
 ('CLEARING', 2085),
 ('DOUGLAS', 4670),
 ('DUNNING', 4568),
 ('EAST GARFIELD PARK', 5337),
 ('EAST SIDE', 5305),
 ('EDGEWATER', 4600),
 ('EDISON PARK', 910),
 ('ENGLEWOOD', 6832),
 ('FOREST GLEN', 1431),
 ('FULLER PARK', 531),
 ('GAGE PARK', 9915),
 ('GARFIELD RIDGE', 4552),
 ('GRAND BOULEVARD', 2809),
 ('GREATER GRAND CROSSING', 4051),
 ('HEGEWISCH', 963),
 ('HERMOSA', 3975),
 ('HUMBOLDT PARK', 8620),
 ('HYDE PARK', 1930),
 ('IRVING PARK', 7764),
 ('JEFFERSON PARK', 1755),
 ('KENWOOD', 4287),
 ('LAKE VIEW', 7055),
 ('LINCOLN PARK', 5615),
```

```
('LINCOLN SQUARE', 4132),
('LOGAN SQUARE', 7351),
('LOOP', 871),
('LOWER WEST SIDE', 7257),
('MCKINLEY PARK', 1552),
('MONTCLARE', 1317),
('MORGAN PARK', 3271),
('MOUNT GREENWOOD', 2091),
('NEAR NORTH SIDE', 3362),
('NEAR SOUTH SIDE', 1378),
('NEAR WEST SIDE', 7975),
('NEW CITY', 7922),
('NORTH CENTER', 7541),
('NORTH LAWNDALE', 5146),
('NORTH PARK', 4210),
('NORWOOD PARK', 6469),
('OAKLAND', 140),
('OHARE', 786),
('PORTAGE PARK', 6954),
('PULLMAN', 1620),
('RIVERDALE', 1547),
('ROGERS PARK', 4068),
('ROSELAND', 7020),
('SOUTH CHICAGO', 4043),
('SOUTH DEERING', 1859),
('SOUTH LAWNDALE', 14793),
('SOUTH SHORE', 4543),
('UPTOWN', 4388),
('WASHINGTON HEIGHTS', 4006),
('WASHINGTON PARK', 2648),
('WEST ELSDON', 3700),
('WEST ENGLEWOOD', 5946),
('WEST GARFIELD PARK', 2622),
('WEST LAWN', 4207),
('WEST PULLMAN', 3240),
('WEST RIDGE', 8197),
('WEST TOWN', 9429),
('WOODLAWN', 4206)]
```

Double-click **here** for a hint

Double-click **here** for another hint

Double-click **here** for the solution.

1.2.9 Problem 9

Get the 5 Community Areas with the least total College Enrollment sorted in ascending order

[18]:
```
%sql select Community_Area_Name, sum(College_Enrollment) AS TOTAL_ENROLLMENT \
   from SCHOOLS \
   group by Community_Area_Name \
   order by TOTAL_ENROLLMENT asc \
   fetch first 5 rows only
```

```
 * ibm_db_sa://mxl17625:***@dashdb-txn-sbox-yp-lon02-02.services.eu-
gb.bluemix.net:50000/BLUDB
Done.
```

[18]:
```
[('OAKLAND', 140),
 ('FULLER PARK', 531),
 ('BURNSIDE', 549),
 ('OHARE', 786),
 ('LOOP', 871)]
```

Double-click **here** for a hint

Double-click **here** for the solution.

1.2.10 Problem 10

Get the hardship index for the community area which has College Enrollment of 4638

[ ]:
```
```

Double-click **here** for the solution.

1.2.11 Problem 11

Get the hardship index for the community area which has the highest value for College Enrollment

[ ]:
```
%%sql
select hardship_index
   from chicago_socioeconomic_data CD, schools CPS
   where CD.ca = CPS.community_area_number
      and college_enrollment = ( select max(college_enrollment)
```

Double-click **here** for the solution.

1.3 Summary

In this lab you learned how to work with a real word dataset using SQL and Python. You learned how to query columns with spaces or special characters in their names and with mixed case names. You also used built in database functions and practiced how to sort, limit, and order result sets, as well as used sub-queries and worked with multiple tables. Copyright © 2018 cognitiveclass.ai. This notebook and its source code are released under the terms of the MIT License.