jupyter-labs-eda-sql-edx

April 15, 2023

Assignment: SQL Notebook for Peer Assignment

Estimated time needed: **60** minutes.

0.1 Introduction

Using this Python notebook you will:

1. Understand the Spacex DataSet
2. Load the dataset into the corresponding table in a Db2 database
3. Execute SQL queries to answer assignment questions

0.2 Overview of the DataSet

SpaceX has gained worldwide attention for a series of historic milestones.

It is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars wheras other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage.

Therefore if we can determine if the first stage will land, we can determine the cost of a launch.

This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

This dataset includes a record for each payload carried during a SpaceX mission into outer space.

0.2.1 Download the datasets

This assignment requires you to load the spacex dataset.

In many cases the dataset to be analyzed is available as a .CSV (comma separated values) file, perhaps on the internet. Click on the link below to download and save the dataset (.CSV file):

Spacex DataSet

Navigate to Go to UI screen

- Refer to this insruction in this link for viewing the Go to UI screen.
- Later click on **Data link(below SQL)** in the Go to UI screen and click on **Load Data** tab.
- Later browse for the downloaded spacex file.
- Once done select the schema andload the file.

If you are facing a problem in uploading the dataset (which is a csv file), you can follow the steps below to upload the .sql file instead of the CSV file:

- Download the file Spacex.sql
- Later click on **SQL** in the **Go to UI Screen**.
- Use the **From file** option to browse for the **SQL** file and upload it.
- Once you upload the script,you can use the **Run All** option to run all the queries to insert the data.

[2]:

```
!pip install --force-reinstall ibm_db==3.1.0 ibm_db_sa==0.3.3
!pip install sqlalchemy==1.3.24
!pip uninstall ipython-sql -y
!pip install ipython-sql==0.4.1
```

```
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
  Downloading
SQLAlchemy-2.0.9-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.7
MB)
                           2.7/2.7 MB
32.5 MB/s eta 0:00:0000:0100:01
Collecting importlib-metadata
  Downloading importlib_metadata-6.3.0-py3-none-any.whl (22 kB)
Collecting typing-extensions>=4.2.0
  Downloading typing_extensions-4.5.0-py3-none-any.whl (27 kB)
Collecting greenlet!=0.4.17
  Downloading
greenlet-2.0.2-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (566
kB)
                          566.1/566.1 kB
36.0 MB/s eta 0:00:00
Collecting zipp>=0.5
  Downloading 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 3.11.0
    Uninstalling zipp-3.11.0:
      Successfully uninstalled zipp-3.11.0
  Attempting uninstall: typing-extensions
```

```
    Found existing installation: typing_extensions 4.4.0
    Uninstalling typing_extensions-4.4.0:
      Successfully uninstalled typing_extensions-4.4.0
  Attempting uninstall: importlib-metadata
    Found existing installation: importlib-metadata 4.11.4
    Uninstalling importlib-metadata-4.11.4:
      Successfully uninstalled importlib-metadata-4.11.4
  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 behaviour 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-6.3.0 sqlalchemy-2.0.9 typing-extensions-4.5.0 zipp-3.15.0
Collecting sqlalchemy==1.3.24
  Using cached SQLAlchemy-1.3.24-cp37-cp37m-manylinux2010_x86_64.whl (1.3 MB)
Installing collected packages: sqlalchemy
  Attempting uninstall: sqlalchemy
    Found existing installation: SQLAlchemy 2.0.9
    Uninstalling SQLAlchemy-2.0.9:
      Successfully uninstalled SQLAlchemy-2.0.9
Successfully installed sqlalchemy-1.3.24
Found existing installation: ipython-sql 0.3.9
Uninstalling ipython-sql-0.3.9:
  Successfully uninstalled ipython-sql-0.3.9
Collecting ipython-sql==0.4.1
  Downloading ipython_sql-0.4.1-py3-none-any.whl (21 kB)
Requirement already satisfied: ipython>=1.0 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-
sql==0.4.1) (7.33.0)
Collecting prettytable<1
  Downloading prettytable-0.7.2.zip (28 kB)
  Preparing metadata (setup.py) … done
Requirement already satisfied: ipython-genutils>=0.1.0 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-
sql==0.4.1) (0.2.0)
Requirement already satisfied: six in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-
sql==0.4.1) (1.16.0)
Requirement already satisfied: sqlalchemy>=0.6.7 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-
sql==0.4.1) (1.3.24)
Requirement already satisfied: sqlparse in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-
sql==0.4.1) (0.4.3)
Requirement already satisfied: jedi>=0.16 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
ipython>=1.0->ipython-sql==0.4.1) (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/lib/python3.7/site-packages (from
ipython>=1.0->ipython-sql==0.4.1) (3.0.33)
Requirement already satisfied: pexpect>4.3 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
ipython>=1.0->ipython-sql==0.4.1) (4.8.0)
Requirement already satisfied: pickleshare in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
ipython>=1.0->ipython-sql==0.4.1) (0.7.5)
Requirement already satisfied: traitlets>=4.2 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
ipython>=1.0->ipython-sql==0.4.1) (5.6.0)
Requirement already satisfied: backcall in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
ipython>=1.0->ipython-sql==0.4.1) (0.2.0)
Requirement already satisfied: decorator in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
ipython>=1.0->ipython-sql==0.4.1) (5.1.1)
Requirement already satisfied: pygments in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
ipython>=1.0->ipython-sql==0.4.1) (2.13.0)
Requirement already satisfied: setuptools>=18.5 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
ipython>=1.0->ipython-sql==0.4.1) (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==0.4.1) (0.1.6)
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==0.4.1) (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
pexpect>4.3->ipython>=1.0->ipython-sql==0.4.1) (0.7.0)
Requirement already satisfied: wcwidth in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from prompt-
toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=1.0->ipython-sql==0.4.1) (0.2.5)
Building wheels for collected packages: prettytable
  Building wheel for prettytable (setup.py) … done
  Created wheel for prettytable: filename=prettytable-0.7.2-py3-none-
any.whl size=13695
sha256=0b858018ccad4b1739ab143f1438dfa7c5bd21a1d982dc8a6fbbd6f3cca201db
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/52/1b/94/5a345ad377639
394ad6c23e3d0bb45fff0fab55b67277b8f7c
Successfully built prettytable
Installing collected packages: prettytable, ipython-sql
  Attempting uninstall: prettytable
    Found existing installation: prettytable 3.5.0
    Uninstalling prettytable-3.5.0:
      Successfully uninstalled prettytable-3.5.0
Successfully installed ipython-sql-0.4.1 prettytable-0.7.2
```

0.2.2 Connect to the database

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

```
[3]: %load_ext sql
```

DB2 magic in case of UI service credentials.

- Use the following format.
- Add security=SSL at the end

%sql ibm_db_sa://my-username:my-password@my-hostname:my-port/my-db-name?security=SSL

```
[4]: %sql ibm_db_sa://gtz84780:Qy9pd1bux71ULzp7@54a2f15b-5c0f-46df-8954-7e38e612c2bd.
      ↪c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/bludb?security=SSL
```

0.3 Tasks

Now write and execute SQL queries to solve the assignment tasks.

0.3.1 Task 1

Display the names of the unique launch sites in the space mission

```
[5]: %sql SELECT DISTINCT launch_site from SPACEX
```

```
 * ibm_db_sa://gtz84780:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32733/bludb
Done.
```

```
[5]: [('CCAFS LC-40',), ('CCAFS SLC-40',), ('KSC LC-39A',), ('VAFB SLC-4E',)]
```

0.3.2 Task 2

Display 5 records where launch sites begin with the string 'KSC'

```
[6]: %sql select * FROM spacex WHERE launch_site like '%KSC%' LIMIT 5
```

```
 * ibm_db_sa://gtz84780:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32733/bludb
Done.
```

```
[6]: [(datetime.date(2017, 2, 19), datetime.time(14, 39), 'F9 FT B1031.1', 'KSC
     LC-39A', 'SpaceX CRS-10', 2490, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'Success
     (ground pad)'),
      (datetime.date(2017, 3, 16), datetime.time(6, 0), 'F9 FT B1030', 'KSC LC-39A',
     'EchoStar 23', 5600, 'GTO', 'EchoStar', 'Success', 'No attempt'),
      (datetime.date(2017, 3, 30), datetime.time(22, 27), 'F9 FT  B1021.2', 'KSC
     LC-39A', 'SES-10', 5300, 'GTO', 'SES', 'Success', 'Success (drone ship)'),
      (datetime.date(2017, 5, 1), datetime.time(11, 15), 'F9 FT B1032.1', 'KSC
     LC-39A', 'NROL-76', 5300, 'LEO', 'NRO', 'Success', 'Success (ground pad)'),
      (datetime.date(2017, 5, 15), datetime.time(23, 21), 'F9 FT B1034', 'KSC
     LC-39A', 'Inmarsat-5 F4', 6070, 'GTO', 'Inmarsat', 'Success', 'No attempt')]
```

0.3.3 Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

```
[7]: %sql select sum(payload_mass__kg_) from spacex WHERE customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://gtz84780:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32733/bludb
Done.
```

```
[7]: [(45596,)]
```

0.3.4 Task 4

Display average payload mass carried by booster version F9 v1.1

```
[8]: %sql select avg(payload_mass__kg_) from spacex where booster_version = 'F9 v1.1'
```

```
 * ibm_db_sa://gtz84780:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32733/bludb
Done.
```

```
[8]: [(2928,)]
```

0.3.5 Task 5

List the date where the first successful landing outcome in drone ship was acheived.
Hint:Use min function

```
[9]: %sql select min(date) from spacex where landing__outcome = 'Success (drone␣
      ↪ship)'
```

```
 * ibm_db_sa://gtz84780:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32733/bludb
Done.
```

```
[9]: [(datetime.date(2016, 4, 8),)]
```

0.3.6 Task 6

List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000

```
[10]: %sql select booster_version from spacex where landing__outcome = 'Success␣
       ↪(ground pad)' and payload_mass__kg_ between 4000 and 6000
```

```
 * ibm_db_sa://gtz84780:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32733/bludb
Done.
```

```
[10]: [('F9 FT B1032.1',), ('F9 B4 B1040.1',), ('F9 B4 B1043.1',)]
```

0.3.7 Task 7

List the total number of successful and failure mission outcomes

```
[11]: %sql select mission_outcome, count(*) from spacex group by mission_outcome
```

```
 * ibm_db_sa://gtz84780:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32733/bludb
Done.
```

```
[11]: [('Failure (in flight)', 1),
       ('Success', 99),
       ('Success (payload status unclear)', 1)]
```

0.3.8 Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
[12]: %sql select booster_version from spacex where payload_mass__kg_ = (select␣
       ↪max(payload_mass__kg_) from spacex)
```

```
 * ibm_db_sa://gtz84780:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32733/bludb
Done.
```

```
[12]: [('F9 B5 B1048.4',),
       ('F9 B5 B1049.4',),
       ('F9 B5 B1051.3',),
       ('F9 B5 B1056.4',),
       ('F9 B5 B1048.5',),
       ('F9 B5 B1051.4',),
       ('F9 B5 B1049.5',),
       ('F9 B5 B1060.2',),
       ('F9 B5 B1058.3',),
       ('F9 B5 B1051.6',),
       ('F9 B5 B1060.3',),
       ('F9 B5 B1049.7',)]
```

0.3.9 Task 9

List the records which will display the month names, succesful landing__outcomes in ground pad ,booster versions, launch__site for the months in year 2017

```
[13]: %sql select monthname(date), landing__outcome, booster_version, launch_site␣
       ↪from spacex where date like '%2017%' and landing__outcome = 'Success (ground␣
       ↪pad)'
```

```
 * ibm_db_sa://gtz84780:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32733/bludb
Done.
```

```
[13]: [('February', 'Success (ground pad)', 'F9 FT B1031.1', 'KSC LC-39A'),
       ('May', 'Success (ground pad)', 'F9 FT B1032.1', 'KSC LC-39A'),
       ('June', 'Success (ground pad)', 'F9 FT B1035.1', 'KSC LC-39A'),
       ('August', 'Success (ground pad)', 'F9 B4 B1039.1', 'KSC LC-39A'),
       ('September', 'Success (ground pad)', 'F9 B4 B1040.1', 'KSC LC-39A'),
       ('December', 'Success (ground pad)', 'F9 FT  B1035.2', 'CCAFS SLC-40')]
```

0.3.10 Task 10

Rank the count of successful landing__outcomes between the date 2010-06-04 and 2017-03-20 in descending order.

```
[14]: %sql select landing__outcome, count(*) from spacex where date between␣
       ↪'2010-06-04' and '2017-03-20' and landing__outcome like '%Success%' group by␣
       ↪landing__outcome order by count desc
```

```
 * ibm_db_sa://gtz84780:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0l
qde00.databases.appdomain.cloud:32733/bludb
Done.
```

```
[14]: [('Success (drone ship)', 5), ('Success (ground pad)', 3)]
```

0.3.11 Reference Links

- Hands-on Lab : String Patterns, Sorting and Grouping
- Hands-on Lab: Built-in functions
- Hands-on Lab : Sub-queries and Nested SELECT Statements
- Hands-on Tutorial: Accessing Databases with SQL magic
- Hands-on Lab: Analyzing a real World Data Set

0.4 Author(s)

Lakshmi Holla

0.5 Other Contributors

Rav Ahuja

0.6 Change log

Date	Version	Changed by	Change Description
2021-10-12	0.4	Lakshmi Holla	Changed markdown
2021-08-24	0.3	Lakshmi Holla	Added library update
2021-07-09	0.2	Lakshmi Holla	Changes made in magic sql
2021-05-20	0.1	Lakshmi Holla	Created Initial Version

##