

jupyter-labs-eda-sql-edx

September 23, 2021

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 whereas 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

0.2.2 Store the dataset in database table

it is highly recommended to manually load the table using the database console **LOAD** tool in **DB2**.

Now open the Db2 console, open the LOAD tool, Select / Drag the .CSV file for the dataset, Next create a New Table, and then follow the steps on-screen instructions to load the data. Name the new table as follows:

SPACEXDATASET

Follow these steps while using old DB2 UI which is having Open Console Screen

Note: While loading Spacex dataset, ensure that detect datatypes is disabled. Later click on the pencil icon(edit option).

1. Change the Date Format by manually typing DD-MM-YYYY and timestamp format as DD-MM-YYYY HH:MM:SS
2. Change the PAYLOAD_MASS__KG_ datatype to INTEGER.

Changes to be considered when having DB2 instance with the new UI having Go to UI screen

- Refer to this instruction in this link for viewing the new 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 and load the file.

```
[1]: !pip install sqlalchemy==1.3.9
      !pip install ibm_db_sa
      !pip install ipython-sql
```

```
Requirement already satisfied: sqlalchemy==1.3.9 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (1.3.9)
Requirement already satisfied: ibm_db_sa in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (0.3.3)
Requirement already satisfied: sqlalchemy>=0.7.3 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ibm_db_sa)
(1.3.9)
Requirement already satisfied: ipython-sql in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (0.3.9)
Requirement already satisfied: ipython>=1.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython-
sql) (7.16.1)
Requirement already satisfied: sqlparse in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython-
sql) (0.4.1)
Requirement already satisfied: prettytable in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython-
sql) (2.1.0)
Requirement already satisfied: ipython-genutils>=0.1.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython-
sql) (0.2.0)
Requirement already satisfied: sqlalchemy>=0.6.7 in
```

```

/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython-
sql) (1.3.9)
Requirement already satisfied: six in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython-
sql) (1.15.0)
Requirement already satisfied: decorator in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
ipython>=1.0->ipython-sql) (4.4.2)
Requirement already satisfied: backcall in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
ipython>=1.0->ipython-sql) (0.2.0)
Requirement already satisfied: prompt-toolkit!=3.0.0,!<3.0.1,<3.1.0,>=2.0.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
ipython>=1.0->ipython-sql) (3.0.19)
Requirement already satisfied: pexpect; sys_platform != "win32" in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
ipython>=1.0->ipython-sql) (4.8.0)
Requirement already satisfied: pygments in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
ipython>=1.0->ipython-sql) (2.9.0)
Requirement already satisfied: traitlets>=4.2 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
ipython>=1.0->ipython-sql) (4.3.3)
Requirement already satisfied: jedi>=0.10 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
ipython>=1.0->ipython-sql) (0.17.2)
Requirement already satisfied: pickleshare in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
ipython>=1.0->ipython-sql) (0.7.5)
Requirement already satisfied: setuptools>=18.5 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
ipython>=1.0->ipython-sql) (49.6.0.post20210108)
Requirement already satisfied: wcwidth in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
prettytable->ipython-sql) (0.2.5)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
prettytable->ipython-sql) (4.6.1)
Requirement already satisfied: ptyprocess>=0.5 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from pexpect;
sys_platform != "win32"->ipython>=1.0->ipython-sql) (0.7.0)
Requirement already satisfied: parso<0.8.0,>=0.7.0 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
jedi>=0.10->ipython>=1.0->ipython-sql) (0.7.1)
Requirement already satisfied: zipp>=0.5 in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from importlib-
metadata; python_version < "3.8"->prettytable->ipython-sql) (3.5.0)
Requirement already satisfied: typing-extensions>=3.6.4; python_version < "3.8"

```

```
in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from
importlib-metadata; python_version < "3.8"->prettytable->ipython-sql) (3.10.0.0)
```

0.2.3 Connect to the database

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

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

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

```
%reload_ext sql
```

DB2 magic in case of old UI service credentials.

In the next cell enter your db2 connection string. Recall you created Service Credentials for your Db2 instance before. From the **uri** field of 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://`

in the following format

```
%sql ibm_db_sa://my-username:my-password@my-hostname:my-port/my-db-name
```

DB2 magic in case of new 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
```

```
[7]: %sql ibm_db_sa://qpz37010:6APDTqWWRUQIDSoW@764264db-9824-4b7c-82df-40d1b13897c2.
      ↪bs2io90108kqb1od8lcg.databases.appdomain.cloud:32536/bludb?security=SSL
```

```
[7]: 'Connected: qpz37010@bludb'
```

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

```
[9]: %%sql
      SELECT DISTINCT LAUNCH_SITE FROM QPZ37010.SPACEXTBL3
```

```
* ibm_db_sa://qpz37010:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1o
d8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

```
[9]: [('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'

[12]: `%%sql`

```
SELECT * FROM QPZ37010.SPACEXTBL3
WHERE LAUNCH_SITE LIKE 'KSC%'
LIMIT 5
```

```
* ibm_db_sa://qpz37010:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8l1cg.databases.appdomain.cloud:32536/bludb
Done.
```

```
[12]: [(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)

[18]: `%%sql`

```
SELECT SUM(PAYLOAD_MASS__KG_) AS "NASA (CRS) TOTAL PAYLOAD MASS" FROM QPZ37010.
↳SPACEXTBL3
WHERE CUSTOMER = 'NASA (CRS)'
```

```
* ibm_db_sa://qpz37010:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8l1cg.databases.appdomain.cloud:32536/bludb
Done.
```

```
[18]: [(45596,)]
```

0.3.4 Task 4

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

[19]: `%%sql`

```
SELECT AVG(PAYLOAD_MASS__KG_) AS "NASA (CRS) AVG PAYLOAD MASS" FROM QPZ37010.
↳SPACEXTBL3
WHERE BOOSTER_VERSION = 'F9 v1.1'
```

```
* ibm_db_sa://qpz37010:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8l1cg.databases.appdomain.cloud:32536/bludb
```

Done.

```
[19]: [(2928,)]
```

0.3.5 Task 5

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

```
[20]: %%sql
```

```
SELECT MIN(DATE) AS "Date where succesful landing outcome in drone ship was_
↪acheived" FROM QPZ37010.SPACEXTBL3
WHERE LANDING__OUTCOME = 'Success (drone ship)'
```

```
* ibm_db_sa://qpz37010:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8l1cg.databases.appdomain.cloud:32536/bludb
Done.
```

```
[20]: [(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

```
[21]: %%sql
```

```
SELECT BOOSTER_VERSION FROM QPZ37010.SPACEXTBL3
WHERE LANDING__OUTCOME = 'Success (ground pad)'
AND
PAYLOAD_MASS__KG_ > 4000
AND
PAYLOAD_MASS__KG_ < 6000
```

```
* ibm_db_sa://qpz37010:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8l1cg.databases.appdomain.cloud:32536/bludb
Done.
```

```
[21]: [('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

```
[23]: %%sql
```

```
SELECT COUNT(MISSION_OUTCOME) AS "Total number of successful and failure_
↪mission outcomes" FROM QPZ37010.SPACEXTBL3
WHERE MISSION_OUTCOME IN ('Failure (in flight)', 'Success')
```

```
* ibm_db_sa://qpz37010:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8lclg.databases.appdomain.cloud:32536/bludb
Done.
```

[23]: [(100,)]

0.3.8 Task 8

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

```
[24]: %%sql
SELECT * FROM
(SELECT BOOSTER_VERSION, MAX(PAYLOAD_MASS__KG_) AS MAX_PAYLOAD_MASS FROM
↳QPZ37010.SPACEXTBL3
GROUP BY BOOSTER_VERSION) as mp
ORDER BY MAX_PAYLOAD_MASS DESC
LIMIT 12
```

```
* ibm_db_sa://qpz37010:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8lclg.databases.appdomain.cloud:32536/bludb
Done.
```

```
[24]: [('F9 B5 B1048.4', 15600),
      ('F9 B5 B1060.3', 15600),
      ('F9 B5 B1060.2', 15600),
      ('F9 B5 B1058.3', 15600),
      ('F9 B5 B1056.4', 15600),
      ('F9 B5 B1051.6', 15600),
      ('F9 B5 B1051.4', 15600),
      ('F9 B5 B1051.3', 15600),
      ('F9 B5 B1049.7', 15600),
      ('F9 B5 B1049.5', 15600),
      ('F9 B5 B1049.4', 15600),
      ('F9 B5 B1048.5', 15600)]
```

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

```
[25]: %%sql
SELECT DATE, MONTHNAME(DATE) AS "Month", YEAR(DATE) AS "Year",
↳LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM QPZ37010.SPACEXTBL3
WHERE LANDING__OUTCOME = 'Success (ground pad)'
AND
YEAR(DATE) = 2017
```

```
* ibm_db_sa://qpz37010:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
d8lclg.databases.appdomain.cloud:32536/bludb
Done.
```

```
[25]: [(datetime.date(2017, 2, 19), 'February', 2017, 'Success (ground pad)', 'F9 FT
B1031.1', 'KSC LC-39A'),
(datetime.date(2017, 5, 1), 'May', 2017, 'Success (ground pad)', 'F9 FT
B1032.1', 'KSC LC-39A'),
(datetime.date(2017, 6, 3), 'June', 2017, 'Success (ground pad)', 'F9 FT
B1035.1', 'KSC LC-39A'),
(datetime.date(2017, 8, 14), 'August', 2017, 'Success (ground pad)', 'F9 B4
B1039.1', 'KSC LC-39A'),
(datetime.date(2017, 9, 7), 'September', 2017, 'Success (ground pad)', 'F9 B4
B1040.1', 'KSC LC-39A'),
(datetime.date(2017, 12, 15), 'December', 2017, '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.

```
[ ]:
```

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
2021-08-24	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

##

© IBM Corporation 2021. All rights reserved.