



Assignment: SQL Notebook for Peer Assignment

Estimated time needed: **60** minutes.

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

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.

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](#)

```
In [1]: !pip install sqlalchemy==1.3.9
```

```

Collecting sqlalchemy==1.3.9
  Downloading SQLAlchemy-1.3.9.tar.gz (6.0 MB)
    _____ 6.0/6.0 MB 80.5 MB/s eta 0:00:00:0
0:0100:01
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: sqlalchemy
  Building wheel for sqlalchemy (setup.py) ... done
  Created wheel for sqlalchemy: filename=SQLAlchemy-1.3.9-cp37-cp37m-linux_x86_64.whl size=1159121 sha256=5a5eaaa80430719e0f3aa7e40aad03e91721a96ad0ebee5c513056cd220b536c
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/03/71/13/010faf12246f72dc76b4150e6e599d13a85b4435e06fb9e51f
Successfully built sqlalchemy
Installing collected packages: sqlalchemy
  Attempting uninstall: sqlalchemy
    Found existing installation: SQLAlchemy 1.3.24
    Uninstalling SQLAlchemy-1.3.24:
      Successfully uninstalled SQLAlchemy-1.3.24
Successfully installed sqlalchemy-1.3.9

```

Connect to the database

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

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

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

```
In [8]: import csv, sqlite3

con = sqlite3.connect("my_data1.db")
cur = con.cursor()
```

```
In [9]: !pip install -q pandas==1.1.5
```

```
In [16]: %sql sqlite:///my_data1.db
```

```
Out[16]: 'Connected: @my_data1.db'
```

```
In [17]: import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.c
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False, method="multi")
%sql ALTER TABLE SPACEXTBL RENAME COLUMN 'Landing _Outcome' TO 'Landing_Outcome'

* sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: "'Landing _Outcome'"
[SQL: ALTER TABLE SPACEXTBL RENAME COLUMN 'Landing _Outcome' TO 'Landing_Outcom
e'];]
(Background on this error at: http://sqlalche.me/e/13/e3q8)
```

Note: This below code is added to remove blank rows from table

```
In [18]: %sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null
```

```
* sqlite:///my_data1.db
(sqlite3.OperationalError) table SPACEXTABLE already exists
[SQL: create table SPACEXTABLE as select * from SPACEXTBL where Date is not null]
(Background on this error at: http://sqlalche.me/e/13/e3q8)
```

Tasks

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

Note: If the column names are in mixed case enclose it in double quotes For Example "Landing_Outcome"

Task 1

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

```
In [19]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
```

```
* sqlite:///my_data1.db
Done.
```

```
Out[19]: Launch_Site
```

```
CCAFS LC-40
```

```
CCAFS SLC-40
```

```
KSC LC-39A
```

```
VAFB SLC-4E
```

Task 2

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

```
In [20]: %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

```
* sqlite:///my_data1.db
Done.
```

Out[20]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)
2013-	- - - - -	- - - - -	CCAFS LC-	SpaceX	- - -	LEO

Task 3

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

```
In [21]: %sql SELECT SUM(PAYLOAD_MASS_KG_) AS TOTAL_PAYLOAD FROM SPACEXTBL WHERE PAYLOAD
* sqlite:///my_data1.db
Done.
```

```
Out[21]: TOTAL_PAYLOAD
111268
```

Task 4

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

```
In [22]: %sql SELECT AVG(PAYLOAD_MASS_KG_) AS AVG_PAYLOAD FROM SPACEXTBL WHERE BOOSTER_V
* sqlite:///my_data1.db
Done.
```

```
Out[22]: AVG_PAYLOAD
2928.4
```

Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

Hint: Use min function

```
In [23]: %sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTBL WHERE LANDING_OUTCOME =
```

```
* sqlite:///my_data1.db
Done.
```

Out[23]: **FIRST_SUCCESS_GP**

2015-12-22

Task 6

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

In [24]: `%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS_KG_ BETW`

```
* sqlite:///my_data1.db
Done.
```

Out[24]: **Booster_Version**

F9 FT B1022

F9 FT B1026

F9 FT B1021.2

F9 FT B1031.2

Task 7

List the total number of successful and failure mission outcomes

In [25]: `%sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL GROUP BY MISSION_OUT`

```
* sqlite:///my_data1.db
Done.
```

Out[25]:

Mission_Outcome	QTY
-----------------	-----

Failure (in flight)	1
---------------------	---

Success	98
---------	----

Success	1
---------	---

Success (payload status unclear)	1
----------------------------------	---

Task 8

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

In [28]: `%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS_KG_ = (S`

```
* sqlite:///my_data1.db
Done.
```

Out[28]: **Booster_Version**

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

Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
In [36]: %sql SELECT substr(Date, 6, 2) AS Month, BOOSTER_VERSION, LAUNCH_SITE FROM SPACE
* sqlite:///my_data1.db
Done.
```

Out[36]:

Month	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

```
In [38]: %sql SELECT LANDING_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL WHERE LANDING_OUTCOM
* sqlite:///my_data1.db
Done.
```

Out[38]:

Landing_Outcome	QTY
Success (drone ship)	5
Success (ground pad)	3

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](#)

Author(s)

Lakshmi Holla

Other Contributors

Rav Ahuja

Change log

Date	Version	Changed by	Change Description
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.