



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 77.0 MB/s eta 0:00:00:00:01
0:01
  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=1a7b79930331053eab491ddb69bae960d555da6cd63d45b13bd51645cace20
  5b
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/03/71/13/010faf12246f72dc7
  6b4150e6e599d13a85b4435e06fb9e51f
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 [2]: `%load_ext sql`

In [3]: `import csv, sqlite3`

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

In [4]: `!pip install -q pandas==1.1.5`

In [5]: `%sql sqlite:///my_data1.db`

Out[5]: 'Connected: @my_data1.db'

In [6]: `import pandas as pd`

```
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False, method="multi")
```

/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/pandas/core/generic.py:2882: UserWarning: The spaces in these column names will not be changed. In pandas versions < 0.14, spaces were converted to underscores.
both result in 0.1234 being formatted as 0.12.

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

In [7]: `%sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null`

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

```
Done.
```

```
Out[7]: []
```

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 [8]: %sql select Distinct("Launch_Site") from SPACEXTBL
```

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

```
Done.
```

```
Out[8]:
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Task 2

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

```
In [9]: %sql select "Launch_Site" from SPACEXTBL where "Launch_Site" like "CCA%"
```

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

```
Done.
```

Out[9]: **Launch_Site**

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS SLC-40

CCAFS SLC-40

CCAFS SLC-40

CCAFS SLC-40

CCAFS SLC-40

Task 3

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

```
In [10]: %sql select sum("PAYLOAD_MASS_KG_") as "total payload mass (kg)" from SPACEXTBL wh
* sqlite:///my_data1.db
Done.
```

```
Out[10]: total payload mass (kg)
45596
```

Task 4

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

```
In [11]: %sql select avg("PAYLOAD_MASS_KG_") as "average payload mass (kg)" from SPACEXTBL
* sqlite:///my_data1.db
Done.
```

```
Out[11]: average payload mass (kg)
2928.4
```

Task 5

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

Hint: Use min function

```
In [12]: %sql select distinct("Landing_Outcome") from SPACEXTBL
* sqlite:///my_data1.db
Done.
```

Out[12]: **Landing_Outcome**

Failure (parachute)
No attempt
Uncontrolled (ocean)
Controlled (ocean)
Failure (drone ship)
Precluded (drone ship)
Success (ground pad)
Success (drone ship)
Success
Failure
No attempt

In [13]: `%sql select min("Date") as "first success date" from SPACEXTBL where "Landing_Outcome" = "Success"`
 * sqlite:///my_data1.db
 Done.

Out[13]: **first success date**

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 [14]: `%sql select "Booster_Version" from SPACEXTBL where "PAYLOAD_MASS_KG_" > 4000 and "PAYLOAD_MASS_KG_" < 6000`
 * sqlite:///my_data1.db
 Done.

Out[14]: **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 [15]: `%sql select distinct("Mission_Outcome") from SPACEXTBL`

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

```
Done.
```

```
Out[15]:
```

Mission_Outcome
Success
Failure (in flight)
Success (payload status unclear)
Success

```
In [16]: %sql select distinct("Mission_Outcome"),count("*") from SPACEXTBL group by "Mission
```

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

```
Done.
```

```
Out[16]:
```

Mission_Outcome	count("*")
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 [17]: %%sql select "Booster_Version"
from SPACEXTBL
WHERE "PAYLOAD_MASS_KG_" =
      (select max("PAYLOAD_MASS_KG_")
from SPACEXTBL)
```

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

```
Done.
```


Out[17]: **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, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
In [18]: %sql select substr(Date, 4, 2) as "month","Landing_Outcome","Booster_Version","Laun
* sqlite:///my_data1.db
Done.
```

```
Out[18]: month Landing_Outcome Booster_Version Launch_Site
```

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 [19]: %sql select distinct("Landing_Outcome"),count(*) from SPACEXTBL where "Date" > "04/
* sqlite:///my_data1.db
Done.
```

```
Out[19]: Landing_Outcome count(*)
```

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.