

Assignment: SQL Notebook for Peer Assignment

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 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 49.1 MB/s eta 0:00:00:00:
010:01
  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=1159122 sha256=8bd1e083b40f91d8111506e861131e08e72c8980d4c96fbd957f413f91ab98b2
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/ef/95/ac/c232f83b415900c26553c64266e1a2b2863bc63e7a5d606c7e
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.
```

Tasks

Task 1

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

```
In [16]: %%sql
        SELECT * FROM SPACEXTBL LIMIT 2;

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

Out[16]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO

In [14]: `%%sql`
`SELECT DISTINCT Launch_Site FROM SPACEXTBL;`

* sqlite:///my_data1.db
Done.

Out[14]:

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 [21]: `%%sql`
`SELECT * FROM SPACEXTBL`
`WHERE Launch_Site LIKE 'CCA%' LIMIT 5;`

* sqlite:///my_data1.db
Done.

Out[21]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

Task 3

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

In [25]:

```
%%sql
SELECT SUM(PAYLOAD_MASS_KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';

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

Out[25]:

SUM(PAYLOAD_MASS_KG_)
45596

Task 4

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

In [35]:

```
%%sql
SELECT AVG(PAYLOAD_MASS_KG_) FROM SPACEXTBL
WHERE Booster_Version LIKE 'F9 v1.1';

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

Out[35]:

AVG(PAYLOAD_MASS_KG_)
2928.4

In [26]:

```
%%sql
SELECT * FROM SPACEXTBL LIMIT 2

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

Out[26]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO

Task 5

List the date when the first successful landing outcome in ground pad was achieved.

In [58]:

```
%%sql
SELECT "Date", "Time (UTC)", "Landing_Outcome" FROM SPACEXTBL
WHERE "Landing_Outcome" LIKE '%success%'
LIMIT 1;

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

Out[58]:

Date	Time (UTC)	Landing_Outcome
22-12-2015	01:29:00	Success (ground pad)

In []:

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 [61]:

```
%%sql
SELECT * FROM SPACEXTBL
WHERE "Mission_Outcome" = 'Success'
AND "PAYLOAD_MASS_KG_" BETWEEN 4000 AND 6000 LIMIT 3;

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

Out[61]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	M
05-08-2014	08:00:00	F9 v1.1	CCAFS LC-40	AsiaSat 8	4535	GTO	AsiaSat	
07-09-2014	05:00:00	F9 v1.1 B1011	CCAFS LC-40	AsiaSat 6	4428	GTO	AsiaSat	
02-03-2015	03:50:00	F9 v1.1 B1014	CCAFS LC-40	ABS-3A Eutelsat 115 West B	4159	GTO	ABS Eutelsat	

Task 7

List the total number of successful and failure mission outcomes

```
In [83]: %%sql
SELECT DISTINCT Mission_Outcome, COUNT(Mission_Outcome) FROM SPACEXTBL
GROUP BY Mission_Outcome;
```

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

```
Out[83]:
```

Mission_Outcome	COUNT(Mission_Outcome)
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 [91]: %%sql
SELECT Booster_Version, PAYLOAD_MASS_KG_ FROM SPACEXTBL
WHERE PAYLOAD_MASS_KG_ = (SELECT MAX(PAYLOAD_MASS_KG_) FROM SPACEXTBL);
```

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

```
Out[91]:
```

Booster_Version	PAYLOAD_MASS_KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

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.

```
In [101... %%sql
```

```
SELECT substr("Date", 4, 2) AS 'Month', "Landing_Outcome", "Booster_Version", "Launch_Site"
WHERE substr("Date",7,4)='2015'
AND substr("Date", 4, 2)
AND "Landing_Outcome" LIKE "%Failure%";
```

* sqlite:///my_data1.db

Done.

Out[101]:

	Month	Landing_Outcome	Booster_Version	Launch_Site
--	-------	-----------------	-----------------	-------------

	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
--	----	----------------------	---------------	-------------

	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
--	----	----------------------	---------------	-------------

Task 10

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

In [122]...

```
%%sql
SELECT "Date", "Landing_Outcome", COUNT("Landing_Outcome") AS LandingOutcomeCount
FROM SPACEXTBL
WHERE "Date" BETWEEN '04-06-2010' and '20-03-2017'
AND "Landing_Outcome" LIKE '%success%'
GROUP BY "Landing_Outcome"
ORDER BY COUNT("Landing_Outcome");
```

* sqlite:///my_data1.db

Done.

Out[122]:

	Date	Landing_Outcome	LandingOutcomeCount	Booster_Version	Launch_Site
--	------	-----------------	---------------------	-----------------	-------------

	18-07-2016	Success (ground pad)	6	F9 FT B1025.1	CCAFS LC-40
--	------------	----------------------	---	---------------	-------------

	08-04-2016	Success (drone ship)	8	F9 FT B1021.1	CCAFS LC-40
--	------------	----------------------	---	---------------	-------------

	07-08-2018	Success	20	F9 B5 B1046.2	CCAFS SLC-40
--	------------	---------	----	---------------	--------------