# 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 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.

### 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 [17]:
```
!pip install sqlalchemy==1.3.9
```

```
Requirement already satisfied: sqlalchemy==1.3.9 in /home/jupyterlab/conda/envs/pyth
on/lib/python3.7/site-packages (1.3.9)
```

## Connect to the database

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

In [18]:
```
#Please uncomment and execute the code below if you are working locally.

!pip install ipython-sql
```

```
Requirement already satisfied: ipython-sql in /home/jupyterlab/conda/envs/python/li
b/python3.7/site-packages (0.3.9)
Requirement already satisfied: prettytable in /home/jupyterlab/conda/envs/python/li
b/python3.7/site-packages (from ipython-sql) (3.7.0)
Requirement already satisfied: ipython>=1.0 in /home/jupyterlab/conda/envs/python/li
b/python3.7/site-packages (from ipython-sql) (7.33.0)
Requirement already satisfied: sqlalchemy>=0.6.7 in /home/jupyterlab/conda/envs/pyth
on/lib/python3.7/site-packages (from ipython-sql) (1.3.9)
Requirement already satisfied: sqlparse in /home/jupyterlab/conda/envs/python/lib/py
thon3.7/site-packages (from ipython-sql) (0.4.4)
Requirement already satisfied: six in /home/jupyterlab/conda/envs/python/lib/python
3.7/site-packages (from ipython-sql) (1.16.0)
Requirement already satisfied: ipython-genutils>=0.1.0 in /home/jupyterlab/conda/env
s/python/lib/python3.7/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: setuptools>=18.5 in /home/jupyterlab/conda/envs/pytho
n/lib/python3.7/site-packages (from ipython>=1.0->ipython-sql) (67.7.2)
Requirement already satisfied: jedi>=0.16 in /home/jupyterlab/conda/envs/python/lib/
python3.7/site-packages (from ipython>=1.0->ipython-sql) (0.18.2)
Requirement already satisfied: decorator in /home/jupyterlab/conda/envs/python/lib/p
ython3.7/site-packages (from ipython>=1.0->ipython-sql) (5.1.1)
Requirement already satisfied: pickleshare in /home/jupyterlab/conda/envs/python/li
b/python3.7/site-packages (from ipython>=1.0->ipython-sql) (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) (5.9.0)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /hom
e/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipyth
on-sql) (3.0.38)
Requirement already satisfied: pygments in /home/jupyterlab/conda/envs/python/lib/py
thon3.7/site-packages (from ipython>=1.0->ipython-sql) (2.15.1)
Requirement already satisfied: backcall in /home/jupyterlab/conda/envs/python/lib/py
thon3.7/site-packages (from ipython>=1.0->ipython-sql) (0.2.0)
Requirement already satisfied: matplotlib-inline in /home/jupyterlab/conda/envs/pyth
on/lib/python3.7/site-packages (from ipython>=1.0->ipython-sql) (0.1.6)
Requirement already satisfied: pexpect>4.3 in /home/jupyterlab/conda/envs/python/li
b/python3.7/site-packages (from ipython>=1.0->ipython-sql) (4.8.0)
Requirement already satisfied: importlib-metadata in /home/jupyterlab/conda/envs/pyt
hon/lib/python3.7/site-packages (from prettytable->ipython-sql) (4.11.4)
Requirement already satisfied: wcwidth in /home/jupyterlab/conda/envs/python/lib/pyt
hon3.7/site-packages (from prettytable->ipython-sql) (0.2.6)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /home/jupyterlab/conda/envs/py
thon/lib/python3.7/site-packages (from jedi>=0.16->ipython>=1.0->ipython-sql) (0.8.
3)
Requirement already satisfied: ptyprocess>=0.5 in /home/jupyterlab/conda/envs/pytho
n/lib/python3.7/site-packages (from pexpect>4.3->ipython>=1.0->ipython-sql) (0.7.0)
Requirement already satisfied: zipp>=0.5 in /home/jupyterlab/conda/envs/python/lib/p
ython3.7/site-packages (from importlib-metadata->prettytable->ipython-sql) (3.15.0)
Requirement already satisfied: typing-extensions>=3.6.4 in /home/jupyterlab/conda/en
vs/python/lib/python3.7/site-packages (from importlib-metadata->prettytable->ipython
-sql) (4.5.0)
```

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

In [21]:
```
import csv, sqlite3
```

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

In [22]:
```
!pip install -q pandas==1.1.5
```

In [23]:
```
%sql sqlite:///my_data1.db
```

Out[23]:
```
'Connected: @my_data1.db'
```

In [24]:
```
import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.clou
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")
```

In [25]:
```
import pandas as pd
df=pd.read_csv("Spacex.csv")
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False, method="multi")
```

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

In [26]:
```
%sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null0
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) table SPACEXTABLE already exists
[SQL: create table SPACEXTABLE as select * from SPACEXTBL where Date is not null0]
(Background on this error at: http://sqlalche.me/e/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 [18]:
```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
```

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

Out[18]:

| 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 [22]:
```
%sql SELECT* FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

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

Out[22]:

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit |
|---|---|---|---|---|---|---|
| 6/4/2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO |
| 12/8/2010 | 15:43:00 | F9 v1.0 B0004 | CCAFS LC-40 | Dragon demo flight C1, two CubeSats, barrel of Brouere cheese | 0 | LEO (ISS) |
| 5/22/2012 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) |
| 10/8/2012 | 0:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) |
| 3/1/2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) |

## Task 3

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

In [30]:
```
%sql SELECT  SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer='NASA (CRS)'
```

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

Out[30]:

| SUM(PAYLOAD_MASS__KG_) |
|---|
| 45596 |

## Task 4

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

In [32]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version='F9 v1.1'
```

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

Out[32]:

| AVG(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 [35]:
```
%sql SELECT Landing_Outcome, MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome='Succes
```

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

Out[35]:

| Landing_Outcome | MIN(Date) |
|---|---|
| Success (ground pad) | 1/8/2018 |

## 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 [37]:
```
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE Landing_Outcome
```

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

Out[37]:

| Booster_Version | PAYLOAD_MASS__KG_ |
|---|---|
| F9 FT B1022 | 4696 |
| F9 FT B1026 | 4600 |
| F9 FT B1021.2 | 5300 |
| F9 FT B1031.2 | 5200 |

## Task 7

**List the total number of successful and failure mission outcomes**

In [39]:
```
%sql SELECT Mission_Outcome, COUNT(*) Total_Number FROM SPACEXTBL GROUP BY Mission_
```

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

Out[39]:

| Mission_Outcome | Total_Number |
|---|---|
| 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 [42]:
```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PA
```

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

Out[42]:

| 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: SQLLite 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 [39]:
```
%sql SELECT substr(Date, 0, 2) as month, Date, Landing_Outcome, Booster_Version, La
WHERE Landing_Outcome = 'Failure (drone ship)' AND substr(Date, 6, 4)='2015'
```

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

Out[39]:

| month | Date | Landing_Outcome | Booster_Version | Launch_Site |
|---|---|---|---|---|
| 1 | 1/10/2015 | Failure (drone ship) | F9 v1.1 B1012 | CCAFS LC-40 |
| 4 | 4/14/2015 | Failure (drone ship) | 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 [37]:

```
%sql SELECT Landing_Outcome, COUNT(*) AS Count_of_Outcome \
FROM SPACEXTBL \
WHERE Date BETWEEN '06/04/2010' AND '3/20/2017' \
GROUP BY Landing_Outcome \
ORDER BY COUNT(*) DESC
```

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

Out[37]:

| Landing_Outcome | Count_of_Outcome |
|---|---|
| Success | 19 |
| No attempt | 9 |
| Success (ground pad) | 4 |
| Success (drone ship) | 4 |
| Failure | 3 |
| Controlled (ocean) | 3 |
| Failure (drone ship) | 2 |
| Failure (parachute) | 1 |

## 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 |