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 [69]:
```
!pip install sqlalchemy==1.3.9
```

```
Requirement already satisfied: sqlalchemy==1.3.9 in /home/jupyterlab/conda/envs/python/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 [70]:
```
#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/lib/python3.7/site-packages (0.3.9)
Requirement already satisfied: prettytable in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-sql) (3.
7.0)
Requirement already satisfied: ipython>=1.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-sql)
(7.33.0)
Requirement already satisfied: sqlalchemy>=0.6.7 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-sq
l) (1.3.9)
Requirement already satisfied: sqlparse in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-sql) (0.4.
4)
Requirement already satisfied: six in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-sql) (1.16.0)
Requirement already satisfied: ipython-genutils>=0.1.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipyt
hon-sql) (0.2.0)
Requirement already satisfied: setuptools>=18.5 in /home/jupyterlab/conda/envs/python/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->ipy
thon-sql) (0.18.2)
Requirement already satisfied: decorator in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipyt
hon-sql) (5.1.1)
Requirement already satisfied: pickleshare in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ip
ython-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 /home/jupyterlab/conda/envs/python/lib/python3.7/sit
e-packages (from ipython>=1.0->ipython-sql) (3.0.38)
Requirement already satisfied: pygments in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipyth
on-sql) (2.15.1)
Requirement already satisfied: backcall in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipyth
on-sql) (0.2.0)
Requirement already satisfied: matplotlib-inline in /home/jupyterlab/conda/envs/python/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/lib/python3.7/site-packages (from ipython>=1.0->ip
ython-sql) (4.8.0)
Requirement already satisfied: importlib-metadata in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from prettytab
le->ipython-sql) (4.11.4)
Requirement already satisfied: wcwidth in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from prettytable->ipython
-sql) (0.2.6)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /home/jupyterlab/conda/envs/python/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/python/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/python3.7/site-packages (from importlib-metadata
->prettytable->ipython-sql) (3.15.0)
Requirement already satisfied: typing-extensions>=3.6.4 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from imp
ortlib-metadata->prettytable->ipython-sql) (4.5.0)
```

In [71]:
```
%load_ext sql
```

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

In [72]:
```
import csv, sqlite3
import sqlite3
import pandas as pd
print(sqlite3.version)
print(sqlite3.sqlite_version)

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

```
2.6.0
3.42.0
```

In [73]:
```
!pip install -q pandas==1.1.5
```

In [74]:
```
%sql sqlite:///my_data1.db
```

Out[74]:
```
'Connected: @my_data1.db'
```

In [75]:
```
import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data
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 [76]:
```
%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/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 [85]:
```
%sql select dISTINCT (LAUNCH_SITE) from SPACEXTABLE;
```

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

Out[85]:

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 [87]:
```
%sql SELECT LAUNCH_SITE from SPACEXTABLE where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

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

Out[87]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Task 3

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

In [88]:
```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTABLE;
```

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

Out[88]:

payloadmass
619967

Task 4

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

In [89]:
```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

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

Out[89]:

payloadmass
6138.287128712871

Task 5

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

Hint:Use min function

In [91]:
```
%sql select min(DATE) from SPACEXTABLE;
```

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

Out[91]:

min(DATE)
2010-06-04

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 [93]:
```
%sql select BOOSTER_VERSION from SPACEXTBL where LANDING_OUTCOME='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 600
```

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

Out[93]:

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 [95]:
```
%sql select count(MISSION_OUTCOME) as missionoutcomes from SPACEXTABLE GROUP BY MISSION_OUTCOME;
```

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

Out[95]:

missionoutcomes
1
98
1
1

Task 8

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

In [96]:
```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTAB
```

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

Out[96]:

boosterversion
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 [100…

```
%sql SELECT substr(Date, 6,2), MISSION_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTABLE where substr(Date,0,5)='2015';
```

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

Out[100]:

substr(Date, 6,2)	Mission_Outcome	Booster_Version	Launch_Site
01	Success	F9 v1.1 B1012	CCAFS LC-40
02	Success	F9 v1.1 B1013	CCAFS LC-40
03	Success	F9 v1.1 B1014	CCAFS LC-40
04	Success	F9 v1.1 B1015	CCAFS LC-40
04	Success	F9 v1.1 B1016	CCAFS LC-40
06	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	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 [102…

```
%sql SELECT LANDING_OUTCOME FROM SPACEXTABLE WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
```

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

Out[102]:

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

Landing_Outcome
No attempt
No attempt
No attempt
Failure (parachute)
Failure (parachute)

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