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

[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 66.6 MB/s eta 0:00:0000:0100:01
  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=c76dc3c3c83ac2bfc15db279d1553854e8e4b63637b1231b0bf54d38b0314c46
  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

[2]:
```
%load_ext sql
```

[3]:
```
import csv, sqlite3

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

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

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

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

[6]:
```
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/Spacex.csv")
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 wer
e 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

[7]:
```
%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

[9]:
```
%%sql

select distinct(Launch_Site) from SPACEXTABLE order by Launch_Site
```

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

[9]:

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'

[11]:
```
%%sql

select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5
```

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

[11]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Task 3

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

*[13]:
```
%%sql

select sum(PAYLOAD_MASS__KG_) as 'Total Payload Mass', Customer
from SPACEXTABLE
where Customer = 'NASA (CRS)'
```

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

[13]:

Total Payload Mass	Customer
45596	NASA (CRS)

Task 4

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

*[14]:
```
%%sql

select avg(PAYLOAD_MASS__KG_) as 'Average Payload Mass', Booster_Version
from SPACEXTABLE
where Booster_Version like 'F9 v1.1%'
```

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

[14]:

Average Payload Mass	Booster_Version
2534.6666666666665	F9 v1.1 B1003

Task 5

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

Hint:Use min function

[18]:
```
%%sql

select min(Date), Landing_Outcome from SPACEXTABLE
where Landing_Outcome = 'Success (ground pad)'
```

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

[18]:

min(Date)	Landing_Outcome
2015-12-22	Success (ground pad)

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

[10]:
```
%%sql

select Booster_Version, Landing_Outcome, PAYLOAD_MASS__KG_
from SPACEXTABLE
where Landing_Outcome = 'Success (drone ship)'
and PAYLOAD_MASS__KG_ between 4000 and 6000
order by PAYLOAD_MASS__KG_
```

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

[10]:

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1026	Success (drone ship)	4600
F9 FT B1022	Success (drone ship)	4696
F9 FT B1031.2	Success (drone ship)	5200
F9 FT B1021.2	Success (drone ship)	5300

Task 7

List the total number of successful and failure mission outcomes

[49]:
```
%%sql

select count(Mission_Outcome) as 'Count of Mission Outcomes', Mission_Outcome
from SPACEXTABLE
group by Mission_Outcome
```

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

[49]:

Count of Mission Outcomes	Mission_Outcome
1	Failure (in flight)
99	Success
1	Success (payload status unclear)

Task 8

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

[19]:
```
%%sql

select Booster_Version, PAYLOAD_MASS__KG_
from SPACEXTABLE
where PAYLOAD_MASS__KG_ in (select max(PAYLOAD_MASS__KG_)
                            from SPACEXTABLE)
```

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

[19]:

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.

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.

[24]:
```
%%sql

select substr(Date,6,2) as 'Month', substr(Date,0,5) as 'Year', Landing_Outcome, Booster_Version, Launch_Site
from SPACEXTABLE
where Landing_Outcome = 'Failure (drone ship)'
and substr(Date,0,5) = '2015'
```

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

[24]:

Month	Year	Landing_Outcome	Booster_Version	Launch_Site
10	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	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.

[39]:
```
%%sql

select count(*) as 'Count', Landing_Outcome
from SPACEXTABLE
where Date between '2010-06-04' and '2017-03-20'
group by Landing_Outcome
order by count(*) desc
```

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

[39]:

Count	Landing_Outcome
10	No attempt
5	Success (ground pad)
5	Success (drone ship)
5	Failure (drone ship)
3	Controlled (ocean)
2	Uncontrolled (ocean)
1	Precluded (drone ship)
1	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