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 [ ]: %pip install sqlalchemy --upgrade
```

```
Requirement already satisfied: sqlalchemy in /home/josephrs/anaconda3/envs/p
yai/lib/python3.11/site-packages (1.3.9)
Collecting sqlalchemy
  Using cached SQLAlchemy-2.0.31-cp311-cp311-manylinux_2_17_x86_64.manylinux
2014_x86_64.whl.metadata (9.6 kB)
Requirement already satisfied: typing-extensions>=4.6.0 in /home/josephrs/an
aconda3/envs/pyai/lib/python3.11/site-packages (from sqlalchemy) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in /home/josephrs/anaconda3/
envs/pyai/lib/python3.11/site-packages (from sqlalchemy) (3.0.3)
Using cached SQLAlchemy-2.0.31-cp311-cp311-manylinux_2_17_x86_64.manylinux20
14_x86_64.whl (3.2 MB)
Installing collected packages: sqlalchemy
  Attempting uninstall: sqlalchemy
    Found existing installation: SQLAlchemy 1.3.9
    Uninstalling SQLAlchemy-1.3.9:
      Successfully uninstalled SQLAlchemy-1.3.9
Successfully installed sqlalchemy-2.0.31
Note: you may need to restart the kernel to use updated packages.
```

Connect to the database

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

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

        %pip install ipython-sql
```

```
Requirement already satisfied: ipython-sql in /home/josephrs/anaconda3/envs/
pyai/lib/python3.11/site-packages (0.5.0)
Requirement already satisfied: prettytable in /home/josephrs/anaconda3/envs/
pyai/lib/python3.11/site-packages (from ipython-sql) (3.10.0)
Requirement already satisfied: ipython in /home/josephrs/anaconda3/envs/pya
i/lib/python3.11/site-packages (from ipython-sql) (8.20.0)
Requirement already satisfied: sqlalchemy>=2.0 in /home/josephrs/anaconda3/e
nvs/pyai/lib/python3.11/site-packages (from ipython-sql) (2.0.31)
Requirement already satisfied: sqlparse in /home/josephrs/anaconda3/envs/pya
i/lib/python3.11/site-packages (from ipython-sql) (0.5.0)
Requirement already satisfied: six in /home/josephrs/anaconda3/envs/pyai/li
b/python3.11/site-packages (from ipython-sql) (1.16.0)
Requirement already satisfied: ipython-genutils in /home/josephrs/anaconda3/
envs/pyai/lib/python3.11/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: typing-extensions>=4.6.0 in /home/josephrs/an
aconda3/envs/pyai/lib/python3.11/site-packages (from sqlalchemy>=2.0->ipytho
n-sql) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in /home/josephrs/anaconda3/
envs/pyai/lib/python3.11/site-packages (from sqlalchemy>=2.0->ipython-sql)
(3.0.3)
Requirement already satisfied: decorator in /home/josephrs/anaconda3/envs/py
ai/lib/python3.11/site-packages (from ipython->ipython-sql) (5.1.1)
Requirement already satisfied: jedi>=0.16 in /home/josephrs/anaconda3/envs/p
yai/lib/python3.11/site-packages (from ipython->ipython-sql) (0.18.1)
Requirement already satisfied: matplotlib-inline in /home/josephrs/anaconda
3/envs/pyai/lib/python3.11/site-packages (from ipython->ipython-sql) (0.1.6)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in /home/joseph
rs/anaconda3/envs/pyai/lib/python3.11/site-packages (from ipython->ipython-s
ql) (3.0.43)
Requirement already satisfied: pygments>=2.4.0 in /home/josephrs/anaconda3/e
nvs/pyai/lib/python3.11/site-packages (from ipython->ipython-sql) (2.15.1)
Requirement already satisfied: stack-data in /home/josephrs/anaconda3/envs/p
yai/lib/python3.11/site-packages (from ipython->ipython-sql) (0.2.0)
Requirement already satisfied: traitlets>=5 in /home/josephrs/anaconda3/env
s/pyai/lib/python3.11/site-packages (from ipython->ipython-sql) (5.7.1)
Requirement already satisfied: pexpect>4.3 in /home/josephrs/anaconda3/envs/
pyai/lib/python3.11/site-packages (from ipython->ipython-sql) (4.8.0)
Requirement already satisfied: wcwidth in /home/josephrs/anaconda3/envs/pya
i/lib/python3.11/site-packages (from prettytable->ipython-sql) (0.2.5)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /home/josephrs/anacond
a3/envs/pyai/lib/python3.11/site-packages (from jedi>=0.16->ipython->ipython
-sql) (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in /home/josephrs/anaconda3/e
nvs/pyai/lib/python3.11/site-packages (from pexpect>4.3->ipython->ipython-sq
l) (0.7.0)
Requirement already satisfied: executing in /home/josephrs/anaconda3/envs/py
ai/lib/python3.11/site-packages (from stack-data->ipython->ipython-sql) (0.
8.3)
Requirement already satisfied: asttokens in /home/josephrs/anaconda3/envs/py
ai/lib/python3.11/site-packages (from stack-data->ipython->ipython-sql) (2.
0.5)
Requirement already satisfied: pure-eval in /home/josephrs/anaconda3/envs/py
ai/lib/python3.11/site-packages (from stack-data->ipython->ipython-sql) (0.
2.2)
Note: you may need to restart the kernel to use updated packages.
```

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

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

In [ ]:
```
import csv, sqlite3

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

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

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

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

Out[ ]:
```
101
```

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

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

```
 * 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: https://sqlalche.me/e/20/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 [ ]:
```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```

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

Out[ ]:

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

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

Out[ ]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)
2013-03-01	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 [ ]:
```
%%sql SELECT SUM(PAYLOAD_MASS__KG_) AS
            Total_Payload_Mass FROM SPACEXTABLE
            WHERE Customer LIKE '%NASA (CRS)%';
```

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

Out[ ]:

Total_Payload_Mass
48213

Task 4

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

In [ ]:
```
%%sql SELECT AVG(PAYLOAD_MASS__KG_) AS
                    Average_Payload_Mass FROM SPACEXTABLE
                    WHERE Booster_Version = 'F9 v1.1';
```

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

Out[ ]:

Average_Payload_Mass
2928.4

Task 5

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

Hint:Use min function

In [ ]:
```
%%sql SELECT MIN(Date) AS
                    First_Successful_Landing_Date FROM SPACEXTABLE
                    WHERE Landing_Outcome = 'Success (ground pad)';
```

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

Out[ ]:

First_Successful_Landing_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 [ ]:
```
%%sql SELECT Booster_Version FROM SPACEXTABLE
                    WHERE Landing_Outcome = 'Success (drone ship)' AND
                    PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

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

Out[ ]:

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 [ ]:
```
%%sql SELECT Mission_Outcome, Landing_Outcome ,COUNT(*) AS Outcome_Count FRO
                    GROUP BY Launch_Site;
```

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

Out[ ]:

Mission_Outcome	Landing_Outcome	Outcome_Count
Success	Failure (parachute)	26
Success	Success (ground pad)	34
Success	Success (ground pad)	25
Success	Uncontrolled (ocean)	16

Task 8

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

In [ ]:
```
%%sql SELECT Booster_Version FROM SPACEXTABLE
                    WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)
```

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

Out[ ]:

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 [ ]:

```
%%sql SELECT substr(Date, 6, 2) AS Month, Booster_Version, Launch_Site,
                    Landing_Outcome FROM SPACEXTABLE WHERE
                    Landing_Outcome = 'Failure (drone ship)' AND
                    substr(Date, 0, 5) = '2015';
```

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

Out[ ]:

Month	Booster_Version	Launch_Site	Landing_Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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 [ ]:

```
%%sql SELECT Landing_Outcome, COUNT(*) AS Outcome_Count FROM SPACEXTABLE
                    WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP E
                    Landing_Outcome ORDER BY Outcome_Count DESC;
```

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

Out[ ]:

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