Skills Network

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

```
!pip install sqlalchemy==1.4
```

```
Collecting sqlalchemy==1.4
  Downloading SQLAlchemy-1.4.0.tar.gz (7.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.4/7.4 MB 120.3 MB/s eta 0:00:00a 0:00:01
  Preparing metadata (setup.py) ... done
Requirement already satisfied: greenlet!=0.4.17 in /opt/conda/lib/python3.11/site-packages (from sqlalchemy==1.4) (3.0.3)
Building wheels for collected packages: sqlalchemy
  Building wheel for sqlalchemy (setup.py) ... done
  Created wheel for sqlalchemy: filename=SQLAlchemy-1.4.0-cp311-cp311-linux_x86_64.whl size=1426270 sha256=7a80354e182894915935a30595ccb734d798957ae0c14cec6c584dde449413b9
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/e7/3e/b3/548935d3cf563dd617440c661265370659ac5bb87086cc1593
Successfully built sqlalchemy
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-1.4.0
```

Connect to the database

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

```
!pip install ipython-sql
!pip install ipython-sql prettytable
```

```
Collecting ipython-sql
  Downloading ipython_sql-0.5.0-py3-none-any.whl.metadata (17 kB)
Collecting prettytable (from ipython-sql)
  Downloading prettytable-3.12.0-py3-none-any.whl.metadata (30 kB)
Requirement already satisfied: ipython in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (8.22.2)
Collecting sqlalchemy>=2.0 (from ipython-sql)
  Downloading SQLAlchemy-2.0.36-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (9.7 kB)
Collecting sqlparse (from ipython-sql)
  Downloading sqlparse-0.5.1-py3-none-any.whl.metadata (3.9 kB)
Requirement already satisfied: six in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (1.16.0)
Requirement already satisfied: ipython-genutils in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: typing-extensions>=4.6.0 in /opt/conda/lib/python3.11/site-packages (from sqlalchemy>=2.0->ipython-sql) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in /opt/conda/lib/python3.11/site-packages (from sqlalchemy>=2.0->ipython-sql) (3.0.3)
Requirement already satisfied: decorator in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (5.1.1)
Requirement already satisfied: jedi>=0.16 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (0.19.1)
Requirement already satisfied: matplotlib-inline in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (0.1.7)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (3.0.42)
Requirement already satisfied: pygments>=2.4.0 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (2.18.0)
Requirement already satisfied: stack-data in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (0.6.2)
Requirement already satisfied: traitlets>=5.13.0 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (5.14.3)
Requirement already satisfied: pexpect>4.3 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (4.9.0)
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.11/site-packages (from prettytable->ipython-sql) (0.2.13)
Requirement already satisfied: parso<0.9.0,>=0.8.3 in /opt/conda/lib/python3.11/site-packages (from jedi>=0.16->ipython->ipython-sql) (0.8.4)
Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/lib/python3.11/site-packages (from pexpect>4.3->ipython->ipython-sql) (0.7.0)
Requirement already satisfied: executing>=1.2.0 in /opt/conda/lib/python3.11/site-packages (from stack-data->ipython->ipython-sql) (2.0.1)
Requirement already satisfied: asttokens>=2.1.0 in /opt/conda/lib/python3.11/site-packages (from stack-data->ipython->ipython-sql) (2.4.1)
Requirement already satisfied: pure-eval in /opt/conda/lib/python3.11/site-packages (from stack-data->ipython->ipython-sql) (0.2.2)
Downloading ipython_sql-0.5.0-py3-none-any.whl (20 kB)
Downloading SQLAlchemy-2.0.36-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.2/3.2 MB 107.4 MB/s eta 0:00:00
Downloading prettytable-3.12.0-py3-none-any.whl (31 kB)
Downloading sqlparse-0.5.1-py3-none-any.whl (44 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.2/44.2 kB 6.5 MB/s eta 0:00:00
```

```
Installing collected packages: sqlparse, sqlalchemy, prettytable, ipython-sql
  Attempting uninstall: sqlalchemy
    Found existing installation: SQLAlchemy 1.4.0
    Uninstalling SQLAlchemy-1.4.0:
      Successfully uninstalled SQLAlchemy-1.4.0
Successfully installed ipython-sql-0.5.0 prettytable-3.12.0 sqlalchemy-2.0.36 sqlparse-0.5.1
Requirement already satisfied: ipython-sql in /opt/conda/lib/python3.11/site-packages (0.5.0)
Requirement already satisfied: prettytable in /opt/conda/lib/python3.11/site-packages (3.12.0)
Requirement already satisfied: ipython in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (8.22.2)
Requirement already satisfied: sqlalchemy>=2.0 in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (2.0.36)
Requirement already satisfied: sqlparse in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (0.5.1)
Requirement already satisfied: six in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (1.16.0)
Requirement already satisfied: ipython-genutils in /opt/conda/lib/python3.11/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.11/site-packages (from prettytable) (0.2.13)
Requirement already satisfied: typing-extensions>=4.6.0 in /opt/conda/lib/python3.11/site-packages (from sqlalchemy>=2.0->ipython-sql) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in /opt/conda/lib/python3.11/site-packages (from sqlalchemy>=2.0->ipython-sql) (3.0.3)
Requirement already satisfied: decorator in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (5.1.1)
Requirement already satisfied: jedi>=0.16 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (0.19.1)
Requirement already satisfied: matplotlib-inline in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (0.1.7)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (3.0.42)
Requirement already satisfied: pygments>=2.4.0 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (2.18.0)
Requirement already satisfied: stack-data in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (0.6.2)
Requirement already satisfied: traitlets>=5.13.0 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (5.14.3)
Requirement already satisfied: pexpect>4.3 in /opt/conda/lib/python3.11/site-packages (from ipython->ipython-sql) (4.9.0)
```

```
%load_ext sql
```

```
import csv, sqlite3
import prettytable
prettytable.DEFAULT = 'DEFAULT'

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

```
!pip install -q pandas
```

```
%sql sqlite:///my_data1.db
```

```
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")
```

```
101
```

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

```
#DROP THE TABLE IF EXISTS

%sql DROP TABLE IF EXISTS SPACEXTABLE;
```

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

```
[]
```

```
%sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null
```

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

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

```
%sql SELECT DISTINCT(Launch_Site) FROM SPACEXTBL
```

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

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A

Task 2

Display 5 records where launch sites begin with the string 'CCA'

```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
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)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt

Task 3

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

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

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

SUM(PAYLOAD_MASS__KG_)

Task 4

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

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

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

AVG(PAYLOAD_MASS__KG_)

Task 5

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

Hint:Use min function

```
%sql SELECT MIN(DATE) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'
```

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

MIN(DATE)

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

```
%%sql
    SELECT Booster_Version FROM SPACEXTBL
    WHERE Landing_Outcome = 'Success (drone ship)'
    AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2

Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Success%'
```

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

COUNT(*)

```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE Landing_Outcome LIKE 'Failure%'
```

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

COUNT(*)

Task 8

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

```
%%sql
    SELECT Booster_Version FROM SPACEXTBL
    WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

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

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.

```
%%sql
    SELECT substr(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site
    FROM SPACEXTBL WHERE substr(Date, 1, 4) = '2015'
    AND Landing_Outcome LIKE 'Failure (drone ship)%';
```

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

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	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.

```
%%sql
    SELECT Landing_Outcome, COUNT(Landing_Outcome) FROM SPACEXTBL
    WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
    GROUP BY Landing_Outcome ORDER BY 2 DESC;
```

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

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

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