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==1.3.9

```

```
Collecting sqlalchemy==1.3.9
  Downloading SQLAlchemy-1.3.9.tar.gz (6.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.0/6.0 MB 35.0 MB/s eta 0:00:00
  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-cp311-cp311-linux_x86_64.whl size=1193594 sha256=e0edc06df45ed89f3
  Stored in directory: /root/.cache/pip/wheels/3a/7c/1e/12404784a68083eb969f877a1808a1847bab897684b56ddc55
Successfully built sqlalchemy
Installing collected packages: sqlalchemy
  Attempting uninstall: sqlalchemy
    Found existing installation: SQLAlchemy 2.0.39
    Uninstalling SQLAlchemy-2.0.39:
      Successfully uninstalled SQLAlchemy-2.0.39
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is
ipython-sql 0.5.0 requires sqlalchemy>=2.0, but you have sqlalchemy 1.3.9 which is incompatible.
langchain 0.3.20 requires SQLAlchemy<3,>=1.4, but you have sqlalchemy 1.3.9 which is incompatible.
Successfully installed sqlalchemy-1.3.9
```

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
```

```
Requirement already satisfied: ipython-sql in /usr/local/lib/python3.11/dist-packages (0.5.0)
Requirement already satisfied: prettytable in /usr/local/lib/python3.11/dist-packages (from ipython-sql) (3.15.1)
Requirement already satisfied: ipython in /usr/local/lib/python3.11/dist-packages (from ipython-sql) (7.34.0)
Collecting sqlalchemy>=2.0 (from ipython-sql)
  Downloading sqlalchemy-2.0.39-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (9.6 kB)
Requirement already satisfied: sqlparse in /usr/local/lib/python3.11/dist-packages (from ipython-sql) (0.5.3)
Requirement already satisfied: six in /usr/local/lib/python3.11/dist-packages (from ipython-sql) (1.17.0)
Requirement already satisfied: ipython-genutils in /usr/local/lib/python3.11/dist-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.11/dist-packages (from sqlalchemy>=2.0->ipython-sq
Requirement already satisfied: typing-extensions>=4.6.0 in /usr/local/lib/python3.11/dist-packages (from sqlalchemy>=2.0->ip
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (75.1
Collecting jedi>=0.16 (from ipython->ipython-sql)
  Downloading jedi-0.19.2-py2.py3-none-any.whl.metadata (22 kB)
Requirement already satisfied: decorator in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (0.7.5)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (5.7.1)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: pygments in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (2.18.0)
Requirement already satisfied: backcall in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (0.2.0)
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (0.1
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (4.9.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.11/dist-packages (from prettytable->ipython-sql) (0.2.13)
Requirement already satisfied: parso<0.9.0,>=0.8.4 in /usr/local/lib/python3.11/dist-packages (from jedi>=0.16->ipython->ipy
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.11/dist-packages (from pexpect>4.3->ipython->ipytho
Downloading sqlalchemy-2.0.39-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.2/3.2 MB 45.7 MB/s eta 0:00:00
Downloading jedi-0.19.2-py2.py3-none-any.whl (1.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.6/1.6 MB 44.2 MB/s eta 0:00:00
Installing collected packages: sqlalchemy, jedi
  Attempting uninstall: sqlalchemy
    Found existing installation: SQLAlchemy 1.3.9
    Uninstalling SQLAlchemy-1.3.9:
      Successfully uninstalled SQLAlchemy-1.3.9
Successfully installed jedi-0.19.2 sqlalchemy-2.0.39
Requirement already satisfied: ipython-sql in /usr/local/lib/python3.11/dist-packages (0.5.0)
Requirement already satisfied: prettytable in /usr/local/lib/python3.11/dist-packages (3.15.1)
Requirement already satisfied: ipython in /usr/local/lib/python3.11/dist-packages (from ipython-sql) (7.34.0)
Requirement already satisfied: sqlalchemy>=2.0 in /usr/local/lib/python3.11/dist-packages (from ipython-sql) (2.0.39)
Requirement already satisfied: sqlparse in /usr/local/lib/python3.11/dist-packages (from ipython-sql) (0.5.3)
Requirement already satisfied: six in /usr/local/lib/python3.11/dist-packages (from ipython-sql) (1.17.0)
Requirement already satisfied: ipython-genutils in /usr/local/lib/python3.11/dist-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.11/dist-packages (from prettytable) (0.2.13)
Requirement already satisfied: greenlet!=0.4.17 in /usr/local/lib/python3.11/dist-packages (from sqlalchemy>=2.0->ipython-sq
Requirement already satisfied: typing-extensions>=4.6.0 in /usr/local/lib/python3.11/dist-packages (from sqlalchemy>=2.0->ip
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (75.1
Requirement already satisfied: jedi>=0.16 in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (0.19.2)
Requirement already satisfied: decorator in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (4.4.2)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (0.7.5)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (5.7.1)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from
Requirement already satisfied: pygments in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (2.18.0)
Requirement already satisfied: backcall in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (0.2.0)
Requirement already satisfied: matplotlib-inline in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (0.1
Requirement already satisfied: pexpect>4.3 in /usr/local/lib/python3.11/dist-packages (from ipython->ipython-sql) (4.9.0)
Requirement already satisfied: parso<0.9.0,>=0.8.4 in /usr/local/lib/python3.11/dist-packages (from jedi>=0.16->ipython->ipy
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.11/dist-packages (from pexpect>4.3->ipython->ipytho
```

```
%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_
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")
```

```
101
```

```
df
```

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
0	2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parach
1	2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parach
2	2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No atte
3	2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No atte
4	2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No atte

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 SPACEXTABLE;
```

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

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'

```
%sql SELECT * FROM SPACEXTABLE 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)

Task 3

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

```
%sql SELECT SUM("PAYLOAD_MASS_KG") FROM SPACEXTABLE WHERE "Customer" = 'NASA (CRS)';
```

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

SUM("PAYLOAD_MASS_KG")
0.0

Task 4

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

```
%sql SELECT AVG("PAYLOAD_MASS_KG") FROM SPACEXTABLE WHERE "Booster_Version" LIKE 'F9 v1.1%';
```

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

AVG("PAYLOAD_MASS_KG")
0.0

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 SPACEXTABLE WHERE "Landing_Outcome" = 'Success' AND "Landing_Outcome" LIKE '%ground pad%';
```

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

MIN("Date")
None

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 SPACEXTABLE WHERE "Landing_Outcome" = 'Success' AND "Landing_Outcome" LIKE '%drone ship%'
```

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

Booster_Version

Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT(*) as Total FROM SPACEXTABLE GROUP BY "Mission_Outcome";
```

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

Mission_Outcome	Total
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

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS_KG" = (SELECT MAX("PAYLOAD_MASS_KG") FROM SPACEXTABLE);
```

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

Booster_Version
F9 v1.0 B0003
F9 v1.0 B0004
F9 v1.0 B0005
F9 v1.0 B0006
F9 v1.0 B0007
F9 v1.1 B1003
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1010
F9 v1.1 B1012
F9 v1.1 B1013
F9 v1.1 B1014
F9 v1.1 B1015
F9 v1.1 B1016
F9 v1.1 B1018
F9 FT B1019
F9 v1.1 B1017
F9 FT B1020
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1024
F9 FT B1025.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1031.1
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 FT B1034
F9 FT B1035.1
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1037
F9 B4 B1039.1
F9 FT B1038.1
F9 B4 B1040.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 FT B1035.2
F9 FT B1036.2
F9 B4 B1043.1
F9 FT B1032.2
F9 FT B1038.2
F9 B4 B1044
F9 B4 B1041.2

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 SPACEXTABLE WHERE substr
```

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

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

```
%sql SELECT "Landing_Outcome", COUNT(*) as Count FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' AND ("L
```

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

Landing_Outcome	Count
Failure (drone ship)	5
Success (ground pad)	3

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