

# jupyter-labs-eda-sql-coursera\_sqllite

September 8, 2025

Assignment: SQL Notebook for Peer Assignment

Estimated time needed: **60** minutes.

## 0.1 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

## 0.2 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 whereas 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.

### 0.2.1 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
136.9 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... one
```

```

Building wheels for collected packages: sqlalchemy
  Building wheel for sqlalchemy (setup.py) ...done
  Created wheel for sqlalchemy:
filename=SQLAlchemy-1.3.9-cp312-cp312-linux_x86_64.whl size=1160111
sha256=a55399e57a943ab0ff76230d1b4b9f05c2547c5bc12540051b7ed6dd8224c5d6
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/b3/1c/42/0e26b8d512adc
6bce10ff71a05229366b4ccec641cd3b42111
Successfully built sqlalchemy
Installing collected packages: sqlalchemy
  Attempting uninstall: sqlalchemy
    Found existing installation: SQLAlchemy 2.0.37
    Uninstalling SQLAlchemy-2.0.37:
      Successfully uninstalled SQLAlchemy-2.0.37
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
jupyterhub 5.2.1 requires SQLAlchemy>=1.4.1, but you have sqlalchemy 1.3.9 which
is incompatible.
Successfully installed sqlalchemy-1.3.9

```

## 0.2.2 Connect to the database

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

```
[2]: !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-3.16.0-py3-none-any.whl.metadata (33 kB)
Requirement already satisfied: ipython in /opt/conda/lib/python3.12/site-
packages (from ipython-sql) (8.31.0)
Collecting sqlalchemy>=2.0 (from ipython-sql)
  Downloading sqlalchemy-2.0.43-cp312-cp312-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (9.6 kB)
Collecting sqlparse (from ipython-sql)
  Downloading sqlparse-0.5.3-py3-none-any.whl.metadata (3.9 kB)
Requirement already satisfied: six in /opt/conda/lib/python3.12/site-packages
(from ipython-sql) (1.17.0)
Requirement already satisfied: ipython-genutils in
/opt/conda/lib/python3.12/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: greenlet>=1 in /opt/conda/lib/python3.12/site-
packages (from sqlalchemy>=2.0->ipython-sql) (3.1.1)
Requirement already satisfied: typing-extensions>=4.6.0 in
/opt/conda/lib/python3.12/site-packages (from sqlalchemy>=2.0->ipython-sql)

```

```

(4.12.2)
Requirement already satisfied: decorator in /opt/conda/lib/python3.12/site-
packages (from ipython->ipython-sql) (5.1.1)
Requirement already satisfied: jedi>=0.16 in /opt/conda/lib/python3.12/site-
packages (from ipython->ipython-sql) (0.19.2)
Requirement already satisfied: matplotlib-inline in
/opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (0.1.7)
Requirement already satisfied: pexpect>4.3 in /opt/conda/lib/python3.12/site-
packages (from ipython->ipython-sql) (4.9.0)
Requirement already satisfied: prompt_toolkit<3.1.0,>=3.0.41 in
/opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (3.0.50)
Requirement already satisfied: pygments>=2.4.0 in
/opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (2.19.1)
Requirement already satisfied: stack_data in /opt/conda/lib/python3.12/site-
packages (from ipython->ipython-sql) (0.6.3)
Requirement already satisfied: traitlets>=5.13.0 in
/opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (5.14.3)
Requirement already satisfied: wcwidth in /opt/conda/lib/python3.12/site-
packages (from prettytable->ipython-sql) (0.2.13)
Requirement already satisfied: parso<0.9.0,>=0.8.4 in
/opt/conda/lib/python3.12/site-packages (from jedi>=0.16->ipython->ipython-sql)
(0.8.4)
Requirement already satisfied: ptyprocess>=0.5 in
/opt/conda/lib/python3.12/site-packages (from pexpect>4.3->ipython->ipython-sql)
(0.7.0)
Requirement already satisfied: executing>=1.2.0 in
/opt/conda/lib/python3.12/site-packages (from stack_data->ipython->ipython-sql)
(2.1.0)
Requirement already satisfied: asttokens>=2.1.0 in
/opt/conda/lib/python3.12/site-packages (from stack_data->ipython->ipython-sql)
(3.0.0)
Requirement already satisfied: pure_eval in /opt/conda/lib/python3.12/site-
packages (from stack_data->ipython->ipython-sql) (0.2.3)
Downloading ipython_sql-0.5.0-py3-none-any.whl (20 kB)
Downloading
sqlalchemy-2.0.43-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(3.3 MB)
3.3/3.3 MB
108.1 MB/s eta 0:00:00
Downloading prettytable-3.16.0-py3-none-any.whl (33 kB)
Downloading sqlparse-0.5.3-py3-none-any.whl (44 kB)
Installing collected packages: sqlparse, sqlalchemy, prettytable, ipython-sql
  Attempting uninstall: sqlalchemy
    Found existing installation: SQLAlchemy 1.3.9
    Uninstalling SQLAlchemy-1.3.9:
      Successfully uninstalled SQLAlchemy-1.3.9
Successfully installed ipython-sql-0.5.0 prettytable-3.16.0 sqlalchemy-2.0.43
sqlparse-0.5.3

```

Requirement already satisfied: ipython-sql in /opt/conda/lib/python3.12/site-packages (0.5.0)

Requirement already satisfied: prettytable in /opt/conda/lib/python3.12/site-packages (3.16.0)

Requirement already satisfied: ipython in /opt/conda/lib/python3.12/site-packages (from ipython-sql) (8.31.0)

Requirement already satisfied: sqlalchemy>=2.0 in /opt/conda/lib/python3.12/site-packages (from ipython-sql) (2.0.43)

Requirement already satisfied: sqlparse in /opt/conda/lib/python3.12/site-packages (from ipython-sql) (0.5.3)

Requirement already satisfied: six in /opt/conda/lib/python3.12/site-packages (from ipython-sql) (1.17.0)

Requirement already satisfied: ipython-genutils in /opt/conda/lib/python3.12/site-packages (from ipython-sql) (0.2.0)

Requirement already satisfied: wcwidth in /opt/conda/lib/python3.12/site-packages (from prettytable) (0.2.13)

Requirement already satisfied: greenlet>=1 in /opt/conda/lib/python3.12/site-packages (from sqlalchemy>=2.0->ipython-sql) (3.1.1)

Requirement already satisfied: typing-extensions>=4.6.0 in /opt/conda/lib/python3.12/site-packages (from sqlalchemy>=2.0->ipython-sql) (4.12.2)

Requirement already satisfied: decorator in /opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (5.1.1)

Requirement already satisfied: jedi>=0.16 in /opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (0.19.2)

Requirement already satisfied: matplotlib-inline in /opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (0.1.7)

Requirement already satisfied: pexpect>4.3 in /opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (4.9.0)

Requirement already satisfied: prompt\_toolkit<3.1.0,>=3.0.41 in /opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (3.0.50)

Requirement already satisfied: pygments>=2.4.0 in /opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (2.19.1)

Requirement already satisfied: stack\_data in /opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (0.6.3)

Requirement already satisfied: traitlets>=5.13.0 in /opt/conda/lib/python3.12/site-packages (from ipython->ipython-sql) (5.14.3)

Requirement already satisfied: parso<0.9.0,>=0.8.4 in /opt/conda/lib/python3.12/site-packages (from jedi>=0.16->ipython->ipython-sql) (0.8.4)

Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/lib/python3.12/site-packages (from pexpect>4.3->ipython->ipython-sql) (0.7.0)

Requirement already satisfied: executing>=1.2.0 in /opt/conda/lib/python3.12/site-packages (from stack\_data->ipython->ipython-sql) (2.1.0)

Requirement already satisfied: asttokens>=2.1.0 in /opt/conda/lib/python3.12/site-packages (from stack\_data->ipython->ipython-sql)

(3.0.0)

Requirement already satisfied: pure\_eval in /opt/conda/lib/python3.12/site-packages (from stack\_data->ipython->ipython-sql) (0.2.3)

```
[3]: %load_ext sql
```

```
[4]: import csv, sqlite3
import prettytable
prettytable.DEFAULT = 'DEFAULT'

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

```
[5]: !pip install -q pandas
```

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

```
[7]: 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")
```

```
[7]: 101
```

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

```
[8]: #DROP THE TABLE IF EXISTS

%sql DROP TABLE IF EXISTS SPACEXTABLE;
```

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

```
[8]: []
```

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

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

```
[9]: []
```

### 0.3 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"

### 0.3.1 Task 1

Display the names of the unique launch sites in the space mission

```
[10]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE;
```

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

```
[10]: [('CCAFS LC-40',), ('VAFB SLC-4E',), ('KSC LC-39A',), ('CCAFS SLC-40',)]
```

### 0.3.2 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]: [('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'),
('2013-03-01', '15:10:00', 'F9 v1.0 B0007', 'CCAFS LC-40', 'SpaceX CRS-2',
677, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'No attempt')]
```

### 0.3.3 Task 3

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

```
[12]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer = 'NASA_
↳(CRS)';
```

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

```
[12]: [(45596,)]
```

### 0.3.4 Task 4

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

```
[13]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version = 'F9_
↳v1.1';
```

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

```
[13]: [(2928.4,)]
```

### 0.3.5 Task 5

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

*Hint: Use min function*

```
[14]: %sql SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground_
      ↳pad)';
```

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

```
[14]: [('2015-12-22',)]
```

### 0.3.6 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

```
[15]: %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.
```

```
[15]: [('F9 FT B1022',), ('F9 FT B1026',), ('F9 FT B1021.2',), ('F9 FT B1031.2',)]
```

### 0.3.7 Task 7

List the total number of successful and failure mission outcomes

```
[16]: %sql SELECT Mission_Outcome, COUNT(*) AS Total FROM SPACEXTABLE GROUP BY_
      ↳Mission_Outcome;
```

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

```
[16]: [('Failure (in flight)', 1),
      ('Success', 98),
      ('Success ', 1),
      ('Success (payload status unclear)', 1)]
```

### 0.3.8 Task 8

List all the booster\_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

```
[17]: %sql SELECT Booster_Version FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT_
      ↳MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

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

```
[17]: [('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 ' ,)]
```

### 0.3.9 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: SQLite 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.

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

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

```
[18]: [('01', 'Failure (drone ship)', 'F9 v1.1 B1012', 'CCAFS LC-40'),
      ('04', 'Failure (drone ship)', 'F9 v1.1 B1015', 'CCAFS LC-40')]
```

### 0.3.10 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.

```
[19]: %sql SELECT Landing_Outcome, COUNT(*) AS OutcomeCount FROM SPACEXTABLE WHERE
↳Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY
↳OutcomeCount DESC;
```

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

```
[19]: [('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)]
```



### **0.3.11 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

### **0.4 Author(s)**

Lakshmi Holla

### **0.5 Other Contributors**

Rav Ahuja

##

© IBM Corporation 2021. All rights reserved.