

jupyter-labs-eda-sql-coursera_sqllite

June 22, 2022

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

0.2.2 Store the dataset in database table

it is highly recommended to manually load the table using the database console **LOAD** tool in **DB2**.

Now open the Db2 console, open the LOAD tool, Select / Drag the .CSV file for the dataset, Next create a New Table, and then follow the steps on-screen instructions to load the data. Name the new table as follows:

SPACEXDATASET

Follow these steps while using old DB2 UI which is having Open Console Screen

Note: While loading Spacex dataset, ensure that detect datatypes is disabled. Later click on the pencil icon(edit option).

1. Change the Date Format by manually typing DD-MM-YYYY and timestamp format as DD-MM-YYYY HH:MM:SS
2. Change the PAYLOAD_MASS__KG_ datatype to INTEGER.

Changes to be considered when having DB2 instance with the new UI having Go to UI screen

- Refer to this instruction in this link for viewing the new Go to UI screen.
- Later click on **Data link(below SQL)** in the Go to UI screen and click on **Load Data** tab.
- Later browse for the downloaded spacex file.
- Once done select the schema and load the file.

```
[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
85.4 MB/s eta 0:00:00:00:01
  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=be93cca9ca7ca3fc7813d6269d73bc7ca062fc95d7b620f81887c58c7c009397
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/03/71/13/010faf12246f7
  2dc76b4150e6e599d13a85b4435e06fb9e51f
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
```

0.2.3 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 were converted to underscores.

both result in 0.1234 being formatted as 0.12.

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

```
[7]: %sql select * From SPACEXTBL Limit 5
```

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

```
[7]: [('04-06-2010', '18:45:00', 'F9 v1.0 B0003', 'CCAFS LC-40', 'Dragon Spacecraft
Qualification Unit', 0, 'LEO', 'SpaceX', 'Success', 'Failure (parachute)'),
('08-12-2010', '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)'),
('22-05-2012', '07:44:00', 'F9 v1.0 B0005', 'CCAFS LC-40', 'Dragon demo flight
C2', 525, 'LEO (ISS)', 'NASA (COTS)', 'Success', 'No attempt'),
('08-10-2012', '00:35:00', 'F9 v1.0 B0006', 'CCAFS LC-40', 'SpaceX CRS-1',
500, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'No attempt'),
```

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

```
[8]: %sql select distinct(Launch_Site) From SPACEXTBL
```

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

```
[8]: [('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'

```
[9]: %sql select * From SPACEXTBL Where Launch_Site like 'CCA%' Limit 5
```

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

```
[9]: [('04-06-2010', '18:45:00', 'F9 v1.0 B0003', 'CCAFS LC-40', 'Dragon Spacecraft  
Qualification Unit', 0, 'LEO', 'SpaceX', 'Success', 'Failure (parachute)'),  
( '08-12-2010', '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)'),  
( '22-05-2012', '07:44:00', 'F9 v1.0 B0005', 'CCAFS LC-40', 'Dragon demo flight  
C2', 525, 'LEO (ISS)', 'NASA (COTS)', 'Success', 'No attempt'),  
( '08-10-2012', '00:35:00', 'F9 v1.0 B0006', 'CCAFS LC-40', 'SpaceX CRS-1',  
500, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'No attempt'),  
( '01-03-2013', '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)

```
[10]: %sql select PAYLOAD_MASS_KG_ From SPACEXTBL Where Customer is 'NASA (CRS)'
```

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

```
[10]: [(500,),  
(677,),  
(2296,),  
(2216,),  
(2395,),  
(1898,),  
(1952,),  
(3136,),  
(2257,)]
```

```
(2490,),  
(2708,),  
(3310,),  
(2205,),  
(2647,),  
(2697,),  
(2500,),  
(2495,),  
(2268,),  
(1977,),  
(2972,)]
```

0.3.4 Task 4

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

```
[11]: %%sql select AVG(PAYLOAD_MASS__KG_) From SPACEXTBL  
Where Booster_Version like 'F9 v1.1%'
```

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

```
[11]: [(2534.6666666666665,)]
```

0.3.5 Task 5

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

Hint: Use min function

```
[12]: #Need double quotes around Landing _Outcome due to the space in that column  
↪name:
```

```
[13]: %%sql select MIN(DATE) From SPACEXTBL Where "Landing _Outcome" is 'Success'  
↪(ground pad)'
```

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

```
[13]: [('01-05-2017',)]
```

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

```
[14]: %%sql select Booster_Version From SPACEXTBL  
Where "Landing _Outcome" is 'Success (drone ship)'  
AND PAYLOAD_MASS__KG_ Between 4000 AND 6000
```

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

```
[14]: [('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

```
[15]: %%sql select (Mission_Outcome), count(Mission_Outcome) as Total From SPACEXTBL
Group by Mission_Outcome
```

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

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

0.3.8 Task 8

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

```
[16]: %sql select MAX(PAYLOAD_MASS__KG_) From SPACEXTBL
```

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

```
[16]: [(15600,)]
```

```
[17]: %%sql select Booster_Version, PAYLOAD_MASS__KG_ From SPACEXTBL
Where PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS__KG_) From SPACEXTBL)
```

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

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

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, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
[18]: %sql select distinct("Landing _Outcome") From SPACEXTBL
```

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

```
[18]: [('Failure (parachute)',),  
      ('No attempt',),  
      ('Uncontrolled (ocean)',),  
      ('Controlled (ocean)',),  
      ('Failure (drone ship)',),  
      ('Precluded (drone ship)',),  
      ('Success (ground pad)',),  
      ('Success (drone ship)',),  
      ('Success',),  
      ('Failure',),  
      ('No attempt ',)]
```

```
[19]: %%sql select Date, substr(Date, 4, 2) as Month, Booster_Version, Launch_Site_  
      ↪From SPACEXTBL  
      Where substr(Date,7,4)='2015'  
      AND "Landing _Outcome" = 'Failure (drone ship)'
```

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

```
[19]: [('10-01-2015', '01', 'F9 v1.1 B1012', 'CCAFS LC-40'),  
      ('14-04-2015', '04', 'F9 v1.1 B1015', 'CCAFS LC-40')]
```

0.3.10 Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
[20]: %%sql select Date, substr(Date, 4, 2) as Month, substr(Date,7,4) as Year,_  
      ↪substr(Date,1,2) as Day,  
      Booster_Version, Launch_Site, "Landing _Outcome"  
      From SPACEXTBL  
      Where "Landing _Outcome" like 'Success%'  
      Order by Year desc, Month desc, Day desc
```

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

[20]: [('06-12-2020', '12', '2020', '06', 'F9 B5 B1058.4 ', 'KSC LC-39A', 'Success'),
('25-11-2020', '11', '2020', '25', 'F9 B5 B1049.7 ', 'CCAFS SLC-40',
'Success'),
('21-11-2020', '11', '2020', '21', 'F9 B5B1063.1', 'VAFB SLC-4E', 'Success'),
('16-11-2020', '11', '2020', '16', 'F9 B5B1061.1 ', 'KSC LC-39A', 'Success'),
('05-11-2020', '11', '2020', '05', 'F9 B5B1062.1', 'CCAFS SLC-40', 'Success'),
('24-10-2020', '10', '2020', '24', 'F9 B5 B1060.3', 'CCAFS SLC-40', 'Success'),
('18-10-2020', '10', '2020', '18', 'F9 B5 B1051.6', 'KSC LC-39A', 'Success'),
('06-10-2020', '10', '2020', '06', 'F9 B5 B1058.3 ', 'KSC LC-39A', 'Success'),
('03-09-2020', '09', '2020', '03', 'F9 B5 B1060.2 ', 'KSC LC-39A', 'Success'),
('30-08-2020', '08', '2020', '30', 'F9 B5 B1059.4', 'CCAFS SLC-40', 'Success'),
('18-08-2020', '08', '2020', '18', 'F9 B5 B1049.6', 'CCAFS SLC-40', 'Success'),
('07-08-2020', '08', '2020', '07', 'F9 B5 B1051.5', 'KSC LC-39A', 'Success'),
('20-07-2020', '07', '2020', '20', 'F9 B5 B1058.2 ', 'CCAFS SLC-40',
'Success'),
('30-06-2020', '06', '2020', '30', 'F9 B5B1060.1', 'CCAFS SLC-40', 'Success'),
('13-06-2020', '06', '2020', '13', 'F9 B5 B1059.3', 'CCAFS SLC-40', 'Success'),
('04-06-2020', '06', '2020', '04', 'F9 B5 B1049.5', 'CCAFS SLC-40', 'Success'),
('30-05-2020', '05', '2020', '30', 'F9 B5B1058.1 ', 'KSC LC-39A', 'Success'),
('22-04-2020', '04', '2020', '22', 'F9 B5 B1051.4', 'KSC LC-39A', 'Success'),
('07-03-2020', '03', '2020', '07', 'F9 B5 B1059.2', 'CCAFS SLC-40', 'Success'),
('29-01-2020', '01', '2020', '29', 'F9 B5 B1051.3', 'CCAFS SLC-40', 'Success'),
('07-01-2020', '01', '2020', '07', 'F9 B5 B1049.4', 'CCAFS SLC-40', 'Success'),
('17-12-2019', '12', '2019', '17', 'F9 B5 B1056.3 ', 'CCAFS SLC-40',
'Success'),
('05-12-2019', '12', '2019', '05', 'F9 B5B1059.1', 'CCAFS SLC-40', 'Success'),
('11-11-2019', '11', '2019', '11', 'F9 B5 B1048.4', 'CCAFS SLC-40', 'Success'),
('25-07-2019', '07', '2019', '25', 'F9 B5 B1056.2 ', 'CCAFS SLC-40',
'Success'),
('12-06-2019', '06', '2019', '12', 'F9 B5 B1051.2 ', 'VAFB SLC-4E', 'Success'),
('24-05-2019', '05', '2019', '24', 'F9 B5 B1049.3', 'CCAFS SLC-40', 'Success'),
('04-05-2019', '05', '2019', '04', 'F9 B5B1056.1 ', 'CCAFS SLC-40', 'Success'),
('02-03-2019', '03', '2019', '02', 'F9 B5B1051.1', 'KSC LC-39A', 'Success'),
('22-02-2019', '02', '2019', '22', 'F9 B5 B1048.3', 'CCAFS SLC-40', 'Success'),
('11-01-2019', '01', '2019', '11', 'F9 B5 B1049.2', 'VAFB SLC-4E', 'Success'),
('03-12-2018', '12', '2018', '03', 'F9 B5 B1046.3', 'VAFB SLC-4E', 'Success'),
('15-11-2018', '11', '2018', '15', 'F9 B5 B1047.2', 'KSC LC-39A', 'Success'),
('08-10-2018', '10', '2018', '08', 'F9 B5 B1048.2', 'VAFB SLC-4E', 'Success'),
('10-09-2018', '09', '2018', '10', 'F9 B5B1049.1', 'CCAFS SLC-40', 'Success'),
('07-08-2018', '08', '2018', '07', 'F9 B5 B1046.2', 'CCAFS SLC-40', 'Success'),
('25-07-2018', '07', '2018', '25', 'F9 B5B1048.1', 'VAFB SLC-4E', 'Success'),
('22-07-2018', '07', '2018', '22', 'F9 B5B1047.1', 'CCAFS SLC-40', 'Success'),
('11-05-2018', '05', '2018', '11', 'F9 B5 B1046.1', 'KSC LC-39A', 'Success
(drone ship)'),
('18-04-2018', '04', '2018', '18', 'F9 B4 B1045.1', 'CCAFS SLC-40', 'Success
(drone ship)'),
('08-01-2018', '01', '2018', '08', 'F9 B4 B1043.1', 'CCAFS SLC-40', 'Success


```
(ground pad)'),
('15-12-2017', '12', '2017', '15', 'F9 FT B1035.2', 'CCAFS SLC-40', 'Success
(ground pad)'),
('30-10-2017', '10', '2017', '30', 'F9 B4 B1042.1', 'KSC LC-39A', 'Success
(drone ship)'),
('11-10-2017', '10', '2017', '11', 'F9 FT B1031.2', 'KSC LC-39A', 'Success
(drone ship)'),
('09-10-2017', '10', '2017', '09', 'F9 B4 B1041.1', 'VAFB SLC-4E', 'Success
(drone ship)'),
('07-09-2017', '09', '2017', '07', 'F9 B4 B1040.1', 'KSC LC-39A', 'Success
(ground pad)'),
('24-08-2017', '08', '2017', '24', 'F9 FT B1038.1', 'VAFB SLC-4E', 'Success
(drone ship)'),
('14-08-2017', '08', '2017', '14', 'F9 B4 B1039.1', 'KSC LC-39A', 'Success
(ground pad)'),
('25-06-2017', '06', '2017', '25', 'F9 FT B1036.1', 'VAFB SLC-4E', 'Success
(drone ship)'),
('23-06-2017', '06', '2017', '23', 'F9 FT B1029.2', 'KSC LC-39A', 'Success
(drone ship)'),
('03-06-2017', '06', '2017', '03', 'F9 FT B1035.1', 'KSC LC-39A', 'Success
(ground pad)'),
('01-05-2017', '05', '2017', '01', 'F9 FT B1032.1', 'KSC LC-39A', 'Success
(ground pad)'),
('30-03-2017', '03', '2017', '30', 'F9 FT B1021.2', 'KSC LC-39A', 'Success
(drone ship)'),
('19-02-2017', '02', '2017', '19', 'F9 FT B1031.1', 'KSC LC-39A', 'Success
(ground pad)'),
('14-01-2017', '01', '2017', '14', 'F9 FT B1029.1', 'VAFB SLC-4E', 'Success
(drone ship)'),
('14-08-2016', '08', '2016', '14', 'F9 FT B1026', 'CCAFS LC-40', 'Success
(drone ship)'),
('18-07-2016', '07', '2016', '18', 'F9 FT B1025.1', 'CCAFS LC-40', 'Success
(ground pad)'),
('27-05-2016', '05', '2016', '27', 'F9 FT B1023.1', 'CCAFS LC-40', 'Success
(drone ship)'),
('06-05-2016', '05', '2016', '06', 'F9 FT B1022', 'CCAFS LC-40', 'Success
(drone ship)'),
('08-04-2016', '04', '2016', '08', 'F9 FT B1021.1', 'CCAFS LC-40', 'Success
(drone ship)'),
('22-12-2015', '12', '2015', '22', 'F9 FT B1019', 'CCAFS LC-40', 'Success
(ground pad)')]
```

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

0.6 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

##

© IBM Corporation 2021. All rights reserved.