



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 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.

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 \(https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv\)](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv)

```
In [103]: !pip install sqlalchemy==1.3.9
!pip install ibm_db_sa
!pip install ipython-sql
```

```
Requirement already satisfied: sqlalchemy==1.3.9 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (1.3.9)
Requirement already satisfied: ibm_db_sa in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (0.3.3)
Requirement already satisfied: sqlalchemy>=0.7.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ibm_db_sa) (1.3.9)
Requirement already satisfied: ipython-sql in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (0.3.9)
Requirement already satisfied: prettytable in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-sql) (3.7.0)
Requirement already satisfied: ipython>=1.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-sql) (7.33.0)
Requirement already satisfied: sqlalchemy>=0.6.7 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-sql) (1.3.9)
Requirement already satisfied: sqlparse in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-sql) (0.4.4)
Requirement already satisfied: six in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-sql) (1.16.0)
Requirement already satisfied: ipython-genutils>=0.1.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: setuptools>=18.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipython-sql) (67.7.2)
Requirement already satisfied: jedi>=0.16 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipython-sql) (0.18.2)
Requirement already satisfied: decorator in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipython-sql) (5.1.1)
Requirement already satisfied: pickleshare in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipython-sql) (0.7.5)
Requirement already satisfied: traitlets>=4.2 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipython-sql) (5.9.0)
Requirement already satisfied: prompt-toolkit!=3.0.0,!<3.0.1,<3.1.0,>=2.0.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipython-sql) (3.0.38)
Requirement already satisfied: pygments in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipython-sql) (2.15.1)
Requirement already satisfied: backcall in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipython-sql) (0.2.0)
Requirement already satisfied: matplotlib-inline in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipython-sql) (0.1.6)
Requirement already satisfied: pexpect>4.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from ipython>=1.0->ipython-sql) (4.8.0)
Requirement already satisfied: importlib-metadata in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from prettytable->ipython-sql) (4.11.4)
Requirement already satisfied: wcwidth in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from prettytable->ipython-sql) (0.2.6)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jedi>=0.16->ipython>=1.0->ipython-sql) (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from pexpect>4.3->ipython>=1.0->ipython-sql) (0.7.0)
Requirement already satisfied: zipp>=0.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from importlib-metadata->prettytable->ipython-sql) (3.15.0)
Requirement already satisfied: typing-extensions>=3.6.4 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from importlib-metadata->prettytable->ipython-sql) (4.5.0)
```

Connect to the database

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

```
In [104]: %load_ext sql
```

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

```
In [105]: import csv, sqlite3
```

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

```
In [106]: !pip install -q pandas==1.1.5
```

```
In [107]: %sql sqlite:///my_data1.db
```

```
Out[107]: 'Connected: @my_data1.db'
```

```
In [108]: 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/SPACEXTBL", con, if_exists='replace', index=False, method="multi")
```

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

```
In [110]: %sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null

* 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: http://sqlalche.me/e/e3q8) (http://sqlalche.me/e/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 [111]: %sql select distinct launch_site from SPACEXTABLE;

* sqlite:///my_data1.db
Done.

Out[111]: 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 [112]: %sql select * from SPACEXTABLE where launch_site like 'CCA%' limit 5;

* sqlite:///my_data1.db
Done.

Out[112]:
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Task 3

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

```
In [113]: %sql select sum(payload_mass_kg_) as total_payload_mass from SPACEXTABLE where customer = 'NASA (CRS)';

* sqlite:///my_data1.db
Done.

Out[113]: total_payload_mass
          45596
```

Task 4

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

In [114]: %sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXTABLE where booster_version like '%F9 v1.1%';

* sqlite:///my_data1.db

Done.

Out[114]:

average_payload_mass
2534.6666666666665

Task 5

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

Hint: Use min function

In [115]: %sql select min(date) as first_successful_landing from SPACEXTABLE where landing_outcome = 'Success (ground pad)';

* sqlite:///my_data1.db

Done.

Out[115]:

first_successful_landing
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 [116]: %sql select booster_version from SPACEXTABLE where landing_outcome = 'Success (drone ship)' and payload_mass__kg_ between 4000 and 6000;

* sqlite:///my_data1.db

Done.

Out[116]:

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 [117]: %sql select mission_outcome, count(*) as total_number from SPACEXTABLE group by mission_outcome;

* sqlite:///my_data1.db

Done.

Out[117]:

Mission_Outcome	total_number
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

In [118]: %sql select booster_version from SPACEXTABLE where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTABLE);

* sqlite:///my_data1.db

Done.

Out[118]:

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: 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.

In [119]: Outcome, Booster_Version, Launch_Site from SPACEXTABLE where Landing_Outcome = 'Failure (drone ship)' and substr(Date,7,4)='2015'

<

* sqlite:///my_data1.db

Done.

Out[119]:

Month	Landing_Outcome	Booster_Version	Launch_Site
-------	-----------------	-----------------	-------------

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 [120]: %%sql select landing_outcome, count(*) as count_outcomes from SPACEXTABLE where date between '2010-06-04' and '2017-03-20' group by landing_outcome order by count_outcomes desc;

* sqlite:///my_data1.db

Done.

Out[120]:

Landing_Outcome	count_outcomes
No attempt	10
Success (ground pad)	5
Success (drone ship)	5
Failure (drone ship)	5
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1
Failure (parachute)	1

Reference Links

- Hands-on Lab : String Patterns, Sorting and Grouping (https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Labs_Coursera_V5/labs/Lab%20-%20String%20Patterns%20-%20Sorting%20-%20Grouping/instructional-labs.md.html?origin=www.coursera.org).
- Hands-on Lab: Built-in functions (https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Labs_Coursera_V5/labs/Lab%20-%20Built-in%20functions%20/Hands-on_Lab_Built-in_Functions.md.html?origin=www.coursera.org).

- [Hands-on Lab : Sub-queries and Nested SELECT Statements \(https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Labs_Coursera_V5/labs/Lab%20-%20Sub-queries%20and%20Nested%20SELECTs%20/instructional-labs.md.html?origin=www.coursera.org\)](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Labs_Coursera_V5/labs/Lab%20-%20Sub-queries%20and%20Nested%20SELECTs%20/instructional-labs.md.html?origin=www.coursera.org)
- [Hands-on Tutorial: Accessing Databases with SQL magic \(https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Module%205/DB0201EN-Week3-1-3-SQLmagic.ipynb\)](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Module%205/DB0201EN-Week3-1-3-SQLmagic.ipynb)
- [Hands-on Lab: Analyzing a real World Data Set \(https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Module%205/DB0201EN-Week3-1-3-SQLmagic.ipynb)

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

© IBM Corporation 2021. All rights reserved.