



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

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

Connect to the database

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

```
In [ ]: #Please uncomment and execute the code below if you are working locally.

        !pip install ipython-sql
```

```
In [46]: %load_ext sql
```

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

```
In [47]: import csv, sqlite3

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

```
In [48]: !pip install -q pandas
```

```
In [49]: %sql sqlite:///my_data1.db
```

```
In [50]: 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")
```

```
Out[50]: 101
```

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

```
In [51]: %sql DROP TABLE SPACEXTABLE
        %sql CREATE TABLE SPACEXTABLE as SELECT * from SPACEXTBL WHERE DATE is not NULL

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

```
Out[51]: []
```

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 [52]: %sql SELECT DISTINCT(LAUNCH_SITE) from SPACEXTBL;
```

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

```
Out[52]:  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 [53]: %sql SELECT LAUNCH_SITE from SPACEXTBL WHERE (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

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

```
Out[53]:  Launch_Site
```

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

CCAFS LC-40

Task 3

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

```
In [54]: %sql SELECT sum(PAYLOAD_MASS__KG_) as PAYLOADMASS from SPACEXTBL;
```

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

```
Out[54]:  PAYLOADMASS
```

619967

Task 4

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

```
In [55]: %sql SELECT avg(PAYLOAD_MASS__KG_) as PAYLOADMASS from SPACEXTBL;
```

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

```
Done.
```

```
Out[55]: PAYLOADMASS
```

```
6138.287128712871
```

Task 5

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

Hint: Use min function

```
In [56]: %sql SELECT min(DATE) from SPACEXTBL;
```

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

```
Done.
```

```
Out[56]: min(DATE)
```

```
2010-06-04
```

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 [64]: %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.
```

```
Out[64]: Booster_Version
```

```
F9 FT B1022
```

```
F9 FT B1026
```

```
F9 FT B1021.2
```

```
F9 FT B1031.2
```

```
In [72]: %sql SELECT * from SPACEXTBL LIMIT 1;
```

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

```
Done.
```

Out[72]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Cu
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	

Task 7

List the total number of successful and failure mission outcomes

```
In [74]: %sql SELECT COUNT(Mission_Outcome) as 'Mission Outcome Count', Mission_Outcome
from SPACEXTBL GROUP BY Mission_Outcome;

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

Out[74]:

Mission Outcome Count	Mission_Outcome
1	Failure (in flight)
98	Success
1	Success
1	Success (payload status unclear)

Task 8

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

```
In [76]: %sql select Booster_version as 'Booster Version' from SPACEXTBL where
PAYLOAD_MASS_KG_=(select max(PAYLOAD_MASS_KG_) from SPACEXTBL);

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

Out[76]: **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, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
In [79]: %sql SELECT substr(Date, 6,2) as 'Month', Mission_Outcome, Booster_Version, Launch_Site FROM SPACEXTBL where substr(Date,0,5)='2015';
```

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

Out[79]: **Month Mission_Outcome Booster_Version Launch_Site**

01	Success	F9 v1.1 B1012	CCAFS LC-40
02	Success	F9 v1.1 B1013	CCAFS LC-40
03	Success	F9 v1.1 B1014	CCAFS LC-40
04	Success	F9 v1.1 B1015	CCAFS LC-40
04	Success	F9 v1.1 B1016	CCAFS LC-40
06	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	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.

```
In [82]: %sql SELECT COUNT(Landing_Outcome) as 'Landing Outcomes Count', Landing_Outcome
FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY
Landing_Outcome ORDER BY DATE DESC;
```

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

Out[82]:

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

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

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

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.