eda-sql-coursera_sqllite

April 19, 2023

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 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.

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

[3]:
```
!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
58.7 MB/s eta 0:00:0000:0100:01
```

```
  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=1159122
sha256=5db41ccdb22c06842bfe8fb7f62a991ff8c57e61a75bc1861ff1c587cce18e11
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/ef/95/ac/c232f83b41590
0c26553c64266e1a2b2863bc63e7a5d606c7e
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
```

0.2.2 Connect to the database

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

[4]:
```
%load_ext sql
```

[5]:
```
import csv, sqlite3

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

[6]:
```
!pip install -q pandas==1.1.5
```

[7]:
```
%sql sqlite:///my_data1.db
```

[7]:
```
'Connected: @my_data1.db'
```

[9]:
```
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")
```

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

[19]:
```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
```

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

[19]:
```
[('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'

[21]:
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

[21]:
```
[('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)

[57]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer LIKE 'NASA␣
↪(CRS)';
```

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

[57]:
```
[(45596,)]
```

0.3.4 Task 4

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

[56]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE␣
↪'F9 V1.1%';
```

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

[56]:
```
[(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

[88]:
```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success␣
↪(ground pad)';
```

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

[88]:
```
[('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

[84]:
```
%sql SELECT DISTINCT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_␣
↪BETWEEN 4000 AND 6000 AND "Landing _Outcome" = 'Success (drone ship)';
```

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

[84]:
```
[('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

[72]:
```
%sql SELECT Mission_Outcome, COUNT(*) FROM SPACEXTBL GROUP BY MISSION_OUTCOME
```

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

[72]:
```
[('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

[75]:
```
%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.
```

[75]:
```
[('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: SQLLite 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.

[80]:
```
%%sql
SELECT substr(Date, 4, 2) as month,booster_version,"Landing _Outcome"
from SPACEXTBL where "Landing _Outcome"
='Failure (drone ship)' and substr(Date,7,4)='2015'
```

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

[80]:
```
[('01', 'F9 v1.1 B1012', 'Failure (drone ship)'),
 ('04', 'F9 v1.1 B1015', 'Failure (drone ship)')]
```

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.

[87]:
```
%%sql
SELECT "Landing _Outcome",count("Landing _Outcome")as LANDING_OUTCOME_COUNT,DATE
from SPACEXTBL where substr(Date,7,4) || substr(Date,4,2) || substr(Date,1,2)␣
 ↪between '20100604'
and '20170320'and "Landing _Outcome" like "Success%"
group by "Landing _Outcome" order by count("Landing _Outcome") desc
```

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

```
[87]: [('Success (drone ship)', 5, '08-04-2016'),
       ('Success (ground pad)', 3, '22-12-2015')]
```

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

##