jupyter-labs-eda-sql-coursera_sqllite

April 3, 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

[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
52.7 MB/s eta 0:00:00:00:0100:01
```

```
  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=5c3f3e46c389382b0fd4387a5f9df39ba75127f2eb603460654f5065069dc7bd
  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

[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

[9]:
```
%sql select Unique(LAUNCH_SITE) from SPACEXTBL;
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) near "Unique": syntax error
[SQL: select Unique(LAUNCH_SITE) from SPACEXTBL;]
(Background on this error at: http://sqlalche.me/e/e3q8)
```

0.3.2 Task 2

Display 5 records where launch sites begin with the string 'CCA'

[10]:
```
%sql SELECT LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;
```

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

[10]:
```
[('CCAFS LC-40',),
 ('CCAFS LC-40',),
 ('CCAFS LC-40',),
 ('CCAFS LC-40',),
 ('CCAFS LC-40',)]
```

0.3.3 Task 3

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

[11]:
```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

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

[11]:
```
[(619967,)]
```

0.3.4 Task 4

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

[12]:
```
%sql select avg(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

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

[12]:
```
[(6138.287128712871,)]
```

0.3.5 Task 5

List the date when the first succesful landing outcome in ground pad was acheived. *Hint:Use min function*

[15]:
```
%sql select min(DATE) from SPACEXTBL;
```

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

[15]:
```
[('01-03-2013',)]
```

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

[17]:
```
%sql select BOOSTER_VERSION from SPACEXTBL where LANDING__OUTCOME='Success␣
↪(drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: LANDING__OUTCOME
[SQL: select BOOSTER_VERSION from SPACEXTBL where LANDING__OUTCOME='Success
(drone ship)' and PAYLOAD_MASS__KG_ BETWEEN 4000 and 6000;]
(Background on this error at: http://sqlalche.me/e/e3q8)
```

0.3.7 Task 7

List the total number of successful and failure mission outcomes

[22]:
```
%sql SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM␣
↪SPACEXTBL where EXTRACT(YEAR FROM DATE)='2015';
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) near "FROM": syntax error
[SQL: SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM
SPACEXTBL where EXTRACT(YEAR FROM DATE)='2015';]
(Background on this error at: http://sqlalche.me/e/e3q8)
```

0.3.8 Task 8

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

[21]:
```
%sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where␣
↪PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

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

[21]:
```
[('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: 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.

[20]:
```
%sql SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM␣
↪SPACEXTBL where EXTRACT(YEAR FROM DATE)='2015';
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) near "FROM": syntax error
[SQL: SELECT MONTH(DATE),MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM
SPACEXTBL where EXTRACT(YEAR FROM DATE)='2015';]
(Background on this error at: http://sqlalche.me/e/e3q8)
```

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.

[19]:
```
%sql SELECT LANDING__OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND␣
↪'2017-03-20' ORDER BY DATE DESC;
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: LANDING__OUTCOME
[SQL: SELECT LANDING__OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND
'2017-03-20' ORDER BY DATE DESC;]
(Background on this error at: http://sqlalche.me/e/e3q8)
```

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

##