SQL Notebook for Peer Assignment

June 3, 2024

[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
60.1 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=1159121
sha256=99cac8f6b028ef5f9958d1ddb8f3cde858cbe1b37dcdc095c6c312a6fa77d878
  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
```

[2]:
```
#Connect to the database
#Please uncomment and execute the code below if you are working locally.
#!pip install ipython-sql
%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.
```

[7]:
```
%sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null
```

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

[7]:
```
[]
```

[8]:
```
##Task 1 Display the names of the unique launch sites in the space mission
%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',)]
```

[9]:
```
###Task 2 Display 5 records where launch sites begin with the string 'CCA'
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

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

[9]:
```
[('2010-06-04', '18:45:00', 'F9 v1.0  B0003', 'CCAFS LC-40', 'Dragon Spacecraft
Qualification Unit', 0, 'LEO', 'SpaceX', 'Success', 'Failure (parachute)'),
 ('2010-12-08', '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', '7:44:00', 'F9 v1.0  B0005', 'CCAFS LC-40', 'Dragon demo flight
C2', 525, 'LEO (ISS)', 'NASA (COTS)', 'Success', 'No attempt'),
 ('2012-10-08', '0:35:00', 'F9 v1.0  B0006', 'CCAFS LC-40', 'SpaceX CRS-1', 500,
'LEO (ISS)', 'NASA (CRS)', 'Success', 'No attempt'),
 ('2013-03-01', '15:10:00', 'F9 v1.0  B0007', 'CCAFS LC-40', 'SpaceX CRS-2',
677, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'No attempt')]
```

[10]:
```
###Task 3 Display the total payload mass carried by boosters launched by NASA␣
↪(CRS)
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

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

[10]:
```
[(45596,)]
```

[11]:
```
###Task 4 Display average payload mass carried by booster version F9 v1.1
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9␣
↪v1.1'
```

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

[11]:
```
[(2928.4,)]
```

[12]:
```
###Task 5
##List the date when the first succesful landing outcome in ground pad was␣
↪acheived.##Hint:Use min function
%sql select min(DATE) from SPACEXTBL where Landing_Outcome = 'Success (ground␣
↪pad)'
```

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

[12]:
```
[('2015-12-22',)]
```

[13]:
```
##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
%sql select BOOSTER_VERSION from SPACEXTBL where Landing_Outcome = 'Success␣
↪(drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000
```

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

[13]:
```
[('F9 FT B1022',), ('F9 FT B1026',), ('F9 FT  B1021.2',), ('F9 FT  B1031.2',)]
```

[14]:
```
##Task 7 List the total number of successful and failure mission outcomes
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME =␣
↪'Success' or MISSION_OUTCOME = 'Failure (in flight)'
```

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

[14]:
```
[(99,)]
```

[15]:
```
###Task 8 List the names of the booster_versions which have carried the maximum␣
↪payload mass. Use a subquery
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select␣
↪max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

[15]:
```
[('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 ',)]
```

[16]:
```
##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,␣
 ↪6,2) as month to get the months and substr(Date,0,5)='2015' for year.
%sql select substr(Date, 6,2) as Month, Landing_Outcome, Booster_Version,␣
 ↪Launch_Site   from SPACEXTABLE where Landing_Outcome = 'Failure (drone␣
 ↪ship)' and substr(Date,0,5) = '2015'
```

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

[16]:
```
[('01', 'Failure (drone ship)', 'F9 v1.1 B1012', 'CCAFS LC-40'),
 ('04', 'Failure (drone ship)', 'F9 v1.1 B1015', 'CCAFS LC-40')]
```

[17]:
```
###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.
%sql select * from SPACEXTBL where Landing_Outcome like 'Success%' and (DATE␣
 ↪between '2010-06-04' and '2017-03-20') order by date desc
```

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

[17]:
```
[('2017-02-19', '14:39:00', 'F9 FT B1031.1', 'KSC LC-39A', 'SpaceX CRS-10',
2490, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'Success (ground pad)'),
 ('2017-01-14', '17:54:00', 'F9 FT B1029.1', 'VAFB SLC-4E', 'Iridium NEXT 1',
9600, 'Polar LEO', 'Iridium Communications', 'Success', 'Success (drone ship)'),
 ('2016-08-14', '5:26:00', 'F9 FT B1026', 'CCAFS LC-40', 'JCSAT-16', 4600,
'GTO', 'SKY Perfect JSAT Group', 'Success', 'Success (drone ship)'),
 ('2016-07-18', '4:45:00', 'F9 FT B1025.1', 'CCAFS LC-40', 'SpaceX CRS-9', 2257,
```

```
'LEO (ISS)', 'NASA (CRS)', 'Success', 'Success (ground pad)'),
 ('2016-05-27', '21:39:00', 'F9 FT B1023.1', 'CCAFS LC-40', 'Thaicom 8', 3100,
'GTO', 'Thaicom', 'Success', 'Success (drone ship)'),
 ('2016-05-06', '5:21:00', 'F9 FT B1022', 'CCAFS LC-40', 'JCSAT-14', 4696,
'GTO', 'SKY Perfect JSAT Group', 'Success', 'Success (drone ship)'),
 ('2016-04-08', '20:43:00', 'F9 FT B1021.1', 'CCAFS LC-40', 'SpaceX CRS-8',
3136, 'LEO (ISS)', 'NASA (CRS)', 'Success', 'Success (drone ship)'),
 ('2015-12-22', '1:29:00', 'F9 FT B1019', 'CCAFS LC-40', 'OG2 Mission 2  11
Orbcomm-OG2 satellites', 2034, 'LEO', 'Orbcomm', 'Success', 'Success (ground
pad)')]
```

[ ]: