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 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.

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
```

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

In [ ]:
```
import csv, sqlite3

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

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

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

In [ ]:
```
import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.c
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")
```

Out[ ]:
```
101
```

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

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

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

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

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 [ ]:
```
%sql select * from spacextable limit 1
```

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

Out[ ]:

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

In [ ]:
```
%sql select distinct(Launch_Site) from spacextable
```

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

Out[ ]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

In [ ]:
```
%sql select distinct(Customer) from spacextable
```

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

Out[ ]:

Customer
SpaceX
NASA (COTS) NRO
NASA (COTS)
NASA (CRS)
MDA
SES
Thaicom
Orbcomm
AsiaSat
U.S. Air Force NASA NOAA
ABS Eutelsat
Turkmenistan National Space Agency
NASA (LSP) NOAA CNES
SKY Perfect JSAT Group
Iridium Communications
EchoStar
NRO
Inmarsat
Bulsatcom
Intelsat
NSPO
U.S. Air Force
SES EchoStar
KT Corporation
Northrop Grumman
Hisdesat exactEarth SpaceX
Hispasat NovaWurks
NASA (LSP)
Thales-Alenia/BTRC
Iridium Communications GFZ , NASA
Telesat
Telkom Indonesia
CONAE

Customer
Es hailSat
Spaceflight Industries
USAF
PSN, SpaceIL / IAI
NASA (CCD)
Canadian Space Agency (CSA)
Spacecom
NASA (CRS), Kacific 1
Sky Perfect JSAT, Kacific 1
NASA (CTS)
NASA (CCDev)
SpaceX, Planet Labs
U.S. Space Force
Republic of Korea Army, Spaceflight Industries (BlackSky)
SpaceX, Spaceflight Industries (BlackSky), Planet Labs
SpaceX, Planet Labs, PlanetIQ
CONAE, PlanetIQ, SpaceX
USSF
NASA (CCP)
NASA / NOAA / ESA / EUMETSAT

Task 2

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

In [ ]:
```
%sql select Launch_Site from spacextable where Launch_Site like '%CCA%' limit 5
```

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

Out[ ]:

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 [ ]:
```
%%sql
select Customer, sum(PAYLOAD_MASS__KG_)
from spacextable
where Customer like '%NASA (CRS)%'
group by Customer;
```

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

Out[ ]:

Customer	sum(PAYLOAD_MASS__KG_)
NASA (CRS)	45596
NASA (CRS), Kacific 1	2617

Task 4

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

In [ ]:
```
%sql select distinct(Booster_Version) from spacextable
```

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

Out[ ]:

Booster_Version
F9 v1.0 B0003
F9 v1.0 B0004
F9 v1.0 B0005
F9 v1.0 B0006
F9 v1.0 B0007
F9 v1.1 B1003
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1010
F9 v1.1 B1012
F9 v1.1 B1013
F9 v1.1 B1014
F9 v1.1 B1015
F9 v1.1 B1016
F9 v1.1 B1018
F9 FT B1019
F9 v1.1 B1017
F9 FT B1020
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1024
F9 FT B1025.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1031.1
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 FT B1034
F9 FT B1035.1
F9 FT B1029.2
F9 FT B1036.1

Booster_Version
F9 FT B1037
F9 B4 B1039.1
F9 FT B1038.1
F9 B4 B1040.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 FT B1035.2
F9 FT B1036.2
F9 B4 B1043.1
F9 FT B1032.2
F9 FT B1038.2
F9 B4 B1044
F9 B4 B1041.2
F9 B4 B1039.2
F9 B4 B1045.1
F9 B5 B1046.1
F9 B4 B1043.2
F9 B4 B1040.2
F9 B4 B1045.2
F9 B5B1047.1
F9 B5B1048.1
F9 B5 B1046.2
F9 B5B1049.1
F9 B5 B1048.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5B1050
F9 B5B1054
F9 B5 B1049.2
F9 B5 B1048.3
F9 B5B1051.1
F9 B5B1056.1

Booster_Version
F9 B5 B1049.3
F9 B5 B1051.2
F9 B5 B1056.2
F9 B5 B1047.3
F9 B5 B1048.4
F9 B5B1059.1
F9 B5 B1056.3
F9 B5 B1049.4
F9 B5 B1046.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1059.2
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5B1058.1
F9 B5 B1049.5
F9 B5 B1059.3
F9 B5B1060.1
F9 B5 B1058.2
F9 B5 B1051.5
F9 B5 B1049.6
F9 B5 B1059.4
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5B1062.1
F9 B5B1061.1
F9 B5B1063.1
F9 B5 B1049.7
F9 B5 B1058.4

In [ ]:
```
%%sql
select Booster_Version, avg(PAYLOAD_MASS__KG_)
```

```
from spacextable
where Booster_Version like '%F9 v1.1%'
group by Booster_Version;
```

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

Out[ ]:

Booster_Version	avg(PAYLOAD_MASS__KG_)
F9 v1.1	2928.4
F9 v1.1 B1003	500.0
F9 v1.1 B1010	2216.0
F9 v1.1 B1011	4428.0
F9 v1.1 B1012	2395.0
F9 v1.1 B1013	570.0
F9 v1.1 B1014	4159.0
F9 v1.1 B1015	1898.0
F9 v1.1 B1016	4707.0
F9 v1.1 B1017	553.0
F9 v1.1 B1018	1952.0

Task 5

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

Hint:Use min function

In [ ]:

```
%%sql


select payload_mass__kg_ from SPACEXTable order by payload_mass__kg_ asc LIMIT 1

--select min(payload_mass__kg_) from spacextable
```

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

Out[ ]:

PAYLOAD_MASS__KG_
0

In [ ]:

```
%sql select * from spacextable limit 1
```

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

Out[ ]:

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

In [ ]:
```
%sql select distinct(Landing_Outcome) from spacextable
```

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

Out[ ]:

Landing_Outcome
Failure (parachute)
No attempt
Uncontrolled (ocean)
Controlled (ocean)
Failure (drone ship)
Precluded (drone ship)
Success (ground pad)
Success (drone ship)
Success
Failure
No attempt

In [ ]:
```
%%sql
select Landing_Outcome, min(date) as first_date
from spacextable
where Landing_Outcome like '%Success (ground pad)%'
```

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

Out[ ]:

Landing_Outcome	first_date
Success (ground pad)	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 [ ]:
```
%%sql
select distinct(Booster_Version) as boosters, landing_outcome as outcomes, PAYLO
from spacextable
where landing_outcome like '%Success (drone ship)%' and PAYLOAD_MASS__KG_ betwee
```

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

Out[ ]:

boosters	outcomes	payload_mass
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Task 7

List the total number of successful and failure mission outcomes

In [ ]:
```
%%sql
select landing_outcome as outcome, count(landing_outcome) as attempts
from spacextable
where landing_outcome not like '%No attempt%'
group by landing_outcome
```

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

Out[ ]:

outcome	attempts
Controlled (ocean)	5
Failure	3
Failure (drone ship)	5
Failure (parachute)	2
Precluded (drone ship)	1
Success	38
Success (drone ship)	14
Success (ground pad)	9
Uncontrolled (ocean)	2

Task 8

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

In [ ]:
```
%%sql
SELECT DISTINCT Booster_Version, PAYLOAD_MASS__KG_ as max_payload
FROM spacextable
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM spacextable
);
```

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

Out[ ]:

Booster_Version	max_payload
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

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.

In [ ]:

```
%%sql
/* SELECT substr(Date, 6, 2) AS month,
       landing_outcome,
       Booster_Version,
       Launch_Site
FROM spacextable
WHERE substr(Date, 0, 5) = '2015'
  AND landing_outcome LIKE '%Failure (drone ship)%'; */

SELECT
    CASE
        WHEN substr(Date, 6, 2) = '01' THEN 'January'
        WHEN substr(Date, 6, 2) = '02' THEN 'February'
        WHEN substr(Date, 6, 2) = '03' THEN 'March'
        WHEN substr(Date, 6, 2) = '04' THEN 'April'
        WHEN substr(Date, 6, 2) = '05' THEN 'May'
        WHEN substr(Date, 6, 2) = '06' THEN 'June'
        WHEN substr(Date, 6, 2) = '07' THEN 'July'
        WHEN substr(Date, 6, 2) = '08' THEN 'August'
        WHEN substr(Date, 6, 2) = '09' THEN 'September'
        WHEN substr(Date, 6, 2) = '10' THEN 'October'
        WHEN substr(Date, 6, 2) = '11' THEN 'November'
        WHEN substr(Date, 6, 2) = '12' THEN 'December'
    END AS month_name,
    landing_outcome,
    Booster_Version,
    Launch_Site
```

```
FROM spacextable
WHERE substr(Date, 0, 5) = '2015'
    AND landing_outcome LIKE '%Failure (drone ship)%';
```

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

Out[ ]:

month_name	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	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 [ ]:

```
%%sql

SELECT landing_outcome,
       COUNT(*) as count_of_outcomes
FROM spacextable
WHERE Date BETWEEN '2010-06-04' and '2017-03-20'
GROUP BY landing_outcome
ORDER BY count_of_outcomes DESC;
```

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

Out[ ]:

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

In [ ]:

```
%sql select distinct(launch_site) from spacextable
```

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

Out[ ]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

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