

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

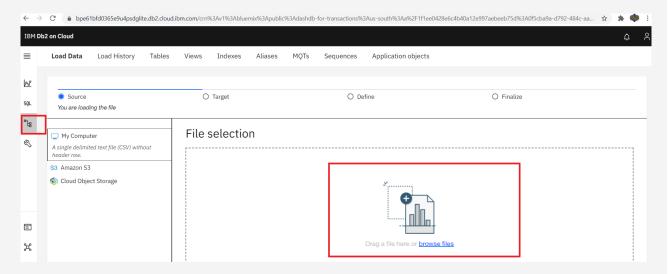
This assignment requires you to load the spacex dataset.

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):

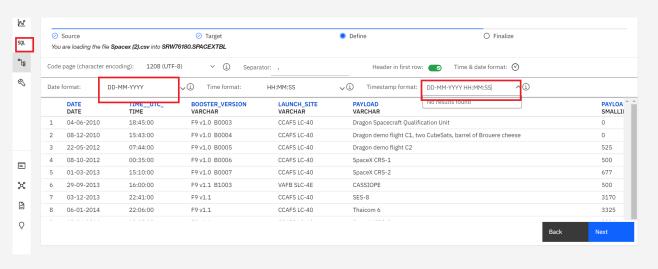
Spacex DataSet

Navigate to the Go to UI screen

- Refer to this insruction in this link for viewing the Go to UI screen.
- Later click on **Data link(below SQL)** in the Go to UI screen and click on **Load Data** tab.
- Later browse for the downloaded spacex file.



Once done select the schema andload the file.



Tn [23]

!pip install sqlalchemy==1.3.9

!pip install ibm_db_sa

!pip install ipython-sql

```
Requirement already satisfied: sqlalchemy==1.3.9 in /opt/conda/envs/Python-3.10/
lib/python3.10/site-packages (1.3.9)
Requirement already satisfied: ibm_db_sa in /opt/conda/envs/Python-3.10/lib/pyth
on3.10/site-packages (0.3.8)
Requirement already satisfied: ibm-db>=2.0.0 in /opt/conda/envs/Python-3.10/lib/
python3.10/site-packages (from ibm_db_sa) (3.1.3)
Requirement already satisfied: sqlalchemy>=0.7.3 in /opt/conda/envs/Python-3.10/
lib/python3.10/site-packages (from ibm_db_sa) (1.3.9)
Requirement already satisfied: ipython-sql in /opt/conda/envs/Python-3.10/lib/py
thon3.10/site-packages (0.4.1)
Requirement already satisfied: prettytable<1 in /opt/conda/envs/Python-3.10/lib/
python3.10/site-packages (from ipython-sql) (0.7.2)
Requirement already satisfied: ipython-genutils>=0.1.0 in /opt/conda/envs/Pythor
-3.10/lib/python3.10/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: sqlparse in /opt/conda/envs/Python-3.10/lib/pythc
n3.10/site-packages (from ipython-sql) (0.4.3)
Requirement already satisfied: six in /opt/conda/envs/Python-3.10/lib/python3.10
/site-packages (from ipython-sql) (1.16.0)
Requirement already satisfied: sqlalchemy>=0.6.7 in /opt/conda/envs/Python-3.10/
lib/python3.10/site-packages (from ipython-sql) (1.3.9)
Requirement already satisfied: ipython>=1.0 in /opt/conda/envs/Python-3.10/lib/r
ython3.10/site-packages (from ipython-sql) (8.4.0)
Requirement already satisfied: jedi>=0.16 in /opt/conda/envs/Python-3.10/lib/pyt
hon3.10/site-packages (from ipython>=1.0->ipython-sql) (0.18.1)
Requirement already satisfied: pygments>=2.4.0 in /opt/conda/envs/Python-3.10/li
b/python3.10/site-packages (from ipython>=1.0->ipython-sql) (2.11.2)
Requirement already satisfied: pickleshare in /opt/conda/envs/Python-3.10/lib/py
thon3.10/site-packages (from ipython>=1.0->ipython-sql) (0.7.5)
Requirement already satisfied: backcall in /opt/conda/envs/Python-3.10/lib/pythc
n3.10/site-packages (from ipython>=1.0->ipython-sql) (0.2.0)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /
opt/conda/envs/Python-3.10/lib/python3.10/site-packages (from ipython>=1.0->ipyt
hon-sql) (3.0.20)
Requirement already satisfied: matplotlib-inline in /opt/conda/envs/Python-3.10/
lib/python3.10/site-packages (from ipython>=1.0->ipython-sql) (0.1.6)
Requirement already satisfied: setuptools>=18.5 in /opt/conda/envs/Python-3.10/l
ib/python3.10/site-packages (from ipython>=1.0->ipython-sql) (63.4.1)
Requirement already satisfied: decorator in /opt/conda/envs/Python-3.10/lib/pyth
on3.10/site-packages (from ipython>=1.0->ipython-sql) (5.1.1)
Requirement already satisfied: pexpect>4.3 in /opt/conda/envs/Python-3.10/lib/py
thon3.10/site-packages (from ipython>=1.0->ipython-sql) (4.8.0)
Requirement already satisfied: traitlets>=5 in /opt/conda/envs/Python-3.10/lib/r
ython3.10/site-packages (from ipython>=1.0->ipython-sql) (5.1.1)
Requirement already satisfied: stack-data in /opt/conda/envs/Python-3.10/lib/pyt
hon3.10/site-packages (from ipython>=1.0->ipython-sql) (0.2.0)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /opt/conda/envs/Python-3.1
0/lib/python3.10/site-packages (from jedi>=0.16->ipython>=1.0->ipython-sql) (0.8
.3)
Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/envs/Python-3.10/li
b/python3.10/site-packages (from pexpect>4.3->ipython>=1.0->ipython-sql) (0.7.0)
Requirement already satisfied: wcwidth in /opt/conda/envs/Python-3.10/lib/pythor
3.10/site-packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=
1.0->ipython-sql) (0.2.5)
Requirement already satisfied: pure-eval in /opt/conda/envs/Python-3.10/lib/pyth
on3.10/site-packages (from stack-data->ipython>=1.0->ipython-sql) (0.2.2)
Requirement already satisfied: executing in /opt/conda/envs/Python-3.10/lib/pyth
on3.10/site-packages (from stack-data->ipython>=1.0->ipython-sql) (0.8.3)
Requirement already satisfied: asttokens in /opt/conda/envs/Python-3.10/lib/pyth
on3.10/site-packages (from stack-data->ipython>=1.0->ipython-sql) (2.0.5)
```

Connect to the database

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

```
In [24]: %reload_ext sql
```

DB2 magic in case of old UI service credentials.

In the next cell enter your db2 connection string. Recall you created Service Credentials for your Db2 instance before. From the **uri** field of your Db2 service credentials copy everything after db2:// (except the double quote at the end) and paste it in the cell below after ibm_db_sa://



in the following format

%sql ibm_db_sa://my-username:my-password@my-hostname:my-port/my-db-name
DB2 magic in case of new UI service credentials.

```
'password'
                  "username": "qdg93144"
                  certificate base64": "LSOtLS1CRUdJTiBDRVJUSUZJOOFURSOtLSOtCk1JSURFakNDOWZxZOF3SUJBZ0lKOVA1SOR3ZTNCTkxiTUEwRONTcl"
FFQkN3VUFNQjR4SERBYUJnTlYKQkFNTUUwbENUU0JEYkc5MVpDQkVZWFJoWW1GelpYTXdIaGNOTWpBd01qSTVNRFF5TVRBeVdoY05NekF3TWpJMgpNRFF5TVF
NUnd3R2dZRFZRUUREQk5KUWswZ1EyeHZkV1FnUkdGMF1XSmhjM1Z6TUlJQk1qQU5CZ2txCmhraUc5dzBCQVFFRkFBT0NBUThBTUlJQkNnS0NBUUVBdXUvbit
DMrTHM3d1dTakxqVE96N3M3M1ZUSU5yYmx3cnRIRulvM1JWTkV6SkNHYW5LSXdZMWZVSUtrCldNM1R0SD15cnFsSGN0Z2pIU1FmRkVTRm1YaHJiODhSQmd0ar
pCaTFBeEVadWNobWZ2QVRmNENOY3EKY21QcHNqdDBPTnIOYnhJMVRyUWxEemNiN1hMSFBrWW91SUprdnVzMUZvaTEySmRNM1MrK31abFZPMUZmZkU3bwpKMjI
GOGŁIUONMSkJvTTFSZ3FPZG90Vm5QOC9EOWZhamNNN0lWd2V4a0lSOTNKR1FJREFRQUJvMU13ClVUQWRCZ05WSFEORUZnUVV1Q3JZanFJQzc1VUpxVmZEMDĎ
UkyRTBUOUt3MlN3RjJ2MXBqaHV4M0lkWWV2SGFVSkRMb0tPd0hSRnFSOHgxZ2dRcGVEcFBnMk5SCkx3R08yek85SWZUMmhLaWd1d2orWnJ5SGxxcHlxQ0pL0lVPekIyWmE2S1YrQTVscEttMWdjV3VHYzMKK1UrVTFzTDd1Ujd3ZFFuVjU0TVU4aERvNi9sVHRMRVB2Mnc3V1NPS1FDK013ejgrTFJMdjVHSW5BN1JySWNhKwc
4ZEttd1pLYThWcnBnMXJ3QzRnY3d1YUhYMUNEWE42K0JIbzhvWG5YWkh6UG91c1dYS1BoaGdXZ2J5CkNDcUdIK0NWNnQ1eFg3b05NS3VNSUNQRVZndnNLWnRc
NVZZbHQ0b1J3dTF1bGdzRDNjek1tbj1LREQKNHB1REFvYTZyMktZZE4xVkxuN3F3VG1TbD1TU05RPT0KLS0tLS1FTkQgQ0VSVE1GSUNBVEUtLS0tLQ0=""" NVZZbHQ0b1J3dTF1bGdzRDNjek1tbj1LREQKNHB1REFvYTZyMktZZE4xVkxuN3F3VG1TbD1TU05RPT0KLS0tLS1FTkQgQ0VSVE1GSUNBVEUtLS0tLQ0=""" NVZZbHQ0b1J3dTF1bGdzRDNjek1tbj1LREQKNHB1REFvYTZyMktZZE4xVkxuN3F3VG1TbD1TU05RPT0KLS0tLS1FTkQgQ0VSVE1GSUNBVEUtLS0tLQ0="" NVZZbHQ0b1J3dTF1bGdzRDNjek1tbj1LREQKNHB1REFvYTZyMktZZE4xVkxuN3F3VG1TbD1TU05RPT0KLS0tLS1FTkQgQ0VSVE1GSUNBVEUtLS0tLQ0=" NVZBHQ0D1TbD1TU05RPT0KLS0tLS0tLS0TB1" NVZBHQ0VSVE1GSUNBVEUTLS0tLQ0=" NVZBHQ0D1TbD1Tu05RPT0KLS0tLS0tLS0TB1" NVZBHQ0VSVE1GSUNBVEUTLS0tLQ0=" NVZBHQ0D1TbD1Tu05RPT0KLS0tLS0tLS0TB1" NVZBHQ0VSVE1GSUNBVEUTLS0tLQ0=" NVZBHQ0D1TbD1Tu05RPT0KLS0tLS0tLS0TB1" NVZBHQ0VSVE1GSUNBVEUTLS0tLQ0=" NVZBHQ0D1TbD1Tu05RPT0KLS0tLS0TB1" NVZBHQ00VSVE1GSUNBVEUTLS0tLQ0=" NVZBHQ0D1TbD1Tu05RPT0KLS0tLS0TB1" NVZBHQ00VSVE1GSUNBVEUTLS0tLQ0=" NVZBHQ00VSVE1GSUNBVEUTLS0TB1" NVZBHQ00VSVE1GSUNBVEUTLS0tLQ0=" NVZBHQ00VSVE1GSUNBVEUTLS0TB1" NVZBHQ00VSVE1GSUNBVEUTLS0tLQ0=" NVZBHQ00VSVE1GSUNBVEUTLS0TB1" NVZBHQ00VSVE1GS
                  "name": "1cbbb1b6-3a1a-4d49-9262-3102a8f7a7c8'
             "composed": [
                                                                                                                                                                                                                atabases.appdomain.c
3/bludb?authSource=admin&replicaSet=replset"
               database": "bludb",
                  'hostname"
                      "port": 32733
```

- Use the following format.
- Add security=SSL at the end

%sql ibm_db_sa://my-username:my-password@my-hostname:my-port/my-db-name?security=SSL

```
In [25]:
         %sql ibm_db_sa://mgb67380:sZHs8AXjMCRJsSvf@98538591-7217-4024-b027-8baa776ffad1.
In [ ]:
In [26]: | %sql ibm_db_sa://
          Tasks
          Now write and execute SQL queries to solve the assignment tasks.
         Task 1
          Display the names of the unique launch sites in the space mission
In [27]: | %sql select Unique(LAUNCH_SITE) from SPACEXTBL;
           * ibm_db_sa://mgb67380:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0ngnrk3
          9u98g.databases.appdomain.cloud:30875/bludb
         Done.
Out[27]:
            launch_site
           CCAFS LC-40
          CCAFS SLC-40
            KSC LC-39A
           VAFB SLC-4E
         Task 2
          Display 5 records where launch sites begin with the string 'CCA'
In [50]: | *sql select LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) like 'CCA%' limit 5;
           * ibm_db_sa://mgb67380:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk3
          9u98g.databases.appdomain.cloud:30875/bludb
         Done.
           launch_site
Out [50]:
          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)
```

sql select sum (PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER='NASA (CRS)'

In [49]:

```
* ibm_db_sa://mgb67380:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk3
         9u98g.databases.appdomain.cloud:30875/bludb
         Done.
Out[49]:
              1
          45596
         Task 4
         Display average payload mass carried by booster version F9 v1.1
In [51]: sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where booster_version like 'F9
          * ibm db sa://mgb67380:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0ngnrk3
         9u98g.databases.appdomain.cloud:30875/bludb
         Done.
Out[51]:
             1
          2534
         Task 5
         List the date when the first successful landing outcome in ground pad was acheived.
         Hint:Use min function
In [59]: sql select min(Date) from SPACEXTBL where LANDING__OUTCOME = 'Success (ground page 1).
          * ibm db sa://mgb67380:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0ngnrk3
         9u98g.databases.appdomain.cloud:30875/bludb
         Done.
                  1
Out [59]:
          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 [53]: sql select distinct BOOSTER VERSION from SPACEXTBL where PAYLOAD MASS KG betw€
           * ibm db sa://mgb67380:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0ngnrk3
         9u98g.databases.appdomain.cloud:30875/bludb
         Done.
Out [53]: booster_version
            F9 FT B1021.2
            F9 FT B1031.2
             F9 FT B1022
             F9 FT B1026
```

Task 7

List the total number of successful and failure mission outcomes

1

In [54]: sql select MISSION_OUTCOME, count(*) from SPACEXTBL group by MISSION_OUTCOME

* ibm_db_sa://mgb67380:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk3
9u98g.databases.appdomain.cloud:30875/bludb
Done.

Out [54]: mission_outcome 2

Failure (in flight) 1

Success 99

Success (payload status unclear)

Task 8

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

In [55]: sql select BOOSTER_VERSION, PAYLOAD_MASS__KG_ from SPACEXTBL where PAYLOAD_MASS_

* ibm_db_sa://mgb67380:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk3
9u98g.databases.appdomain.cloud:30875/bludb
Done.

Out[55]:	booster_version	payload_masskg_

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 failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

In [58]: sql select BOOSTER_VERSION, LAUNCH_SITE from SPACEXTBL where LANDING__OUTCOME='F

* ibm_db_sa://mgb67380:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk3
9u98g.databases.appdomain.cloud:30875/bludb
Done.

Out [58]: booster_version launch_site F9 v1.1 B1012 CCAFS LC-40

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 [57]: sql select LANDING__OUTCOME, count(*) as qty from SPACEXTBL where Date between

* ibm_db_sa://mgb67380:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb

Out [57]: landing_outcome qty

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

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
2021-10-12	0.4	Lakshmi Holla	Changed markdown
2021-08-24	0.3	Lakshmi Holla	Added library update
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.