Assignment: SQL Notebook

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

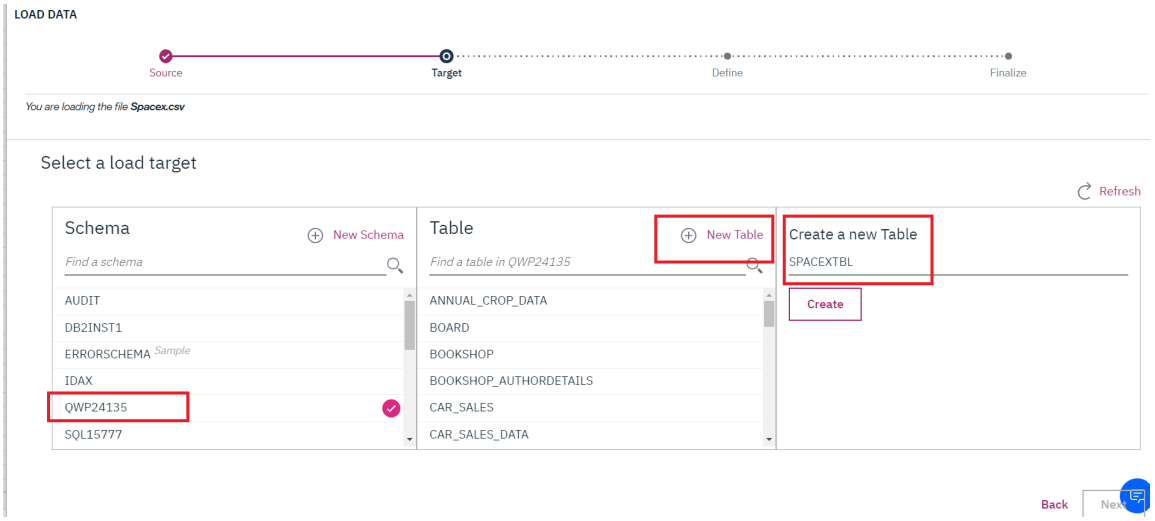
This assignment requires you to load the spacex dataset.

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

Store the dataset in database table

it is highly recommended to manually load the table using the database console LOAD tool in DB2.



Now open the Db2 console, open the LOAD tool, Select / Drag the .CSV file for the dataset, Next create a New Table, and then follow the steps on-screen instructions to load the data. Name the new table as follows:

SPACEXDATASET

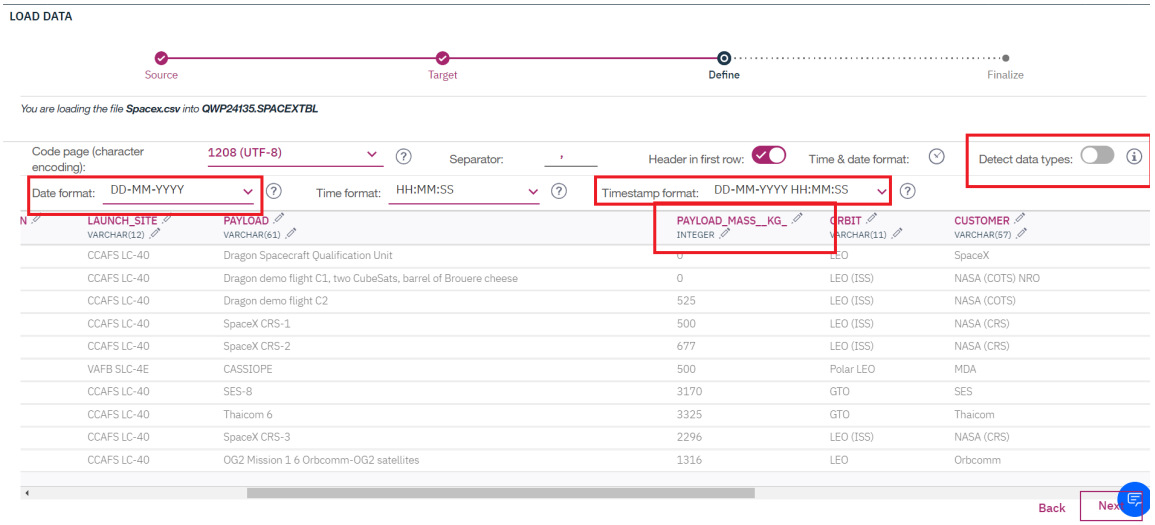
Follow these steps while using old DB2 UI which is having Open Console Screen

Note:While loading Spacex dataset, ensure that detect datatypes is disabled. Later click on the pencil icon(edit option).

1. Change the Date Format by manually typing DD-MM-YYYY and timestamp format as DD-MM-YYYY HH:MM:SS.

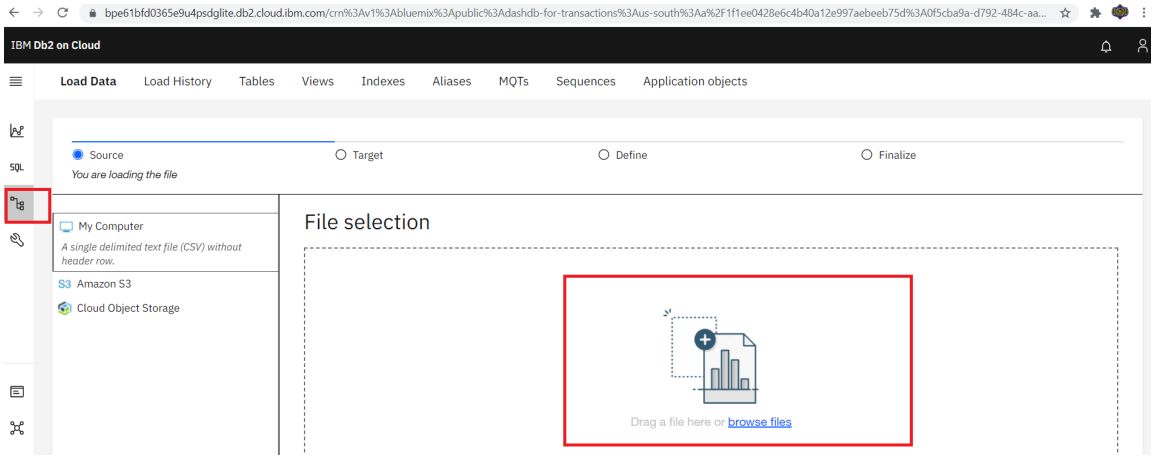
   Here you should place the cursor at Date field and manually type as DD-MM-YYYY.

2. Change the PAYLOAD_MASS__KG_ datatype to INTEGER.

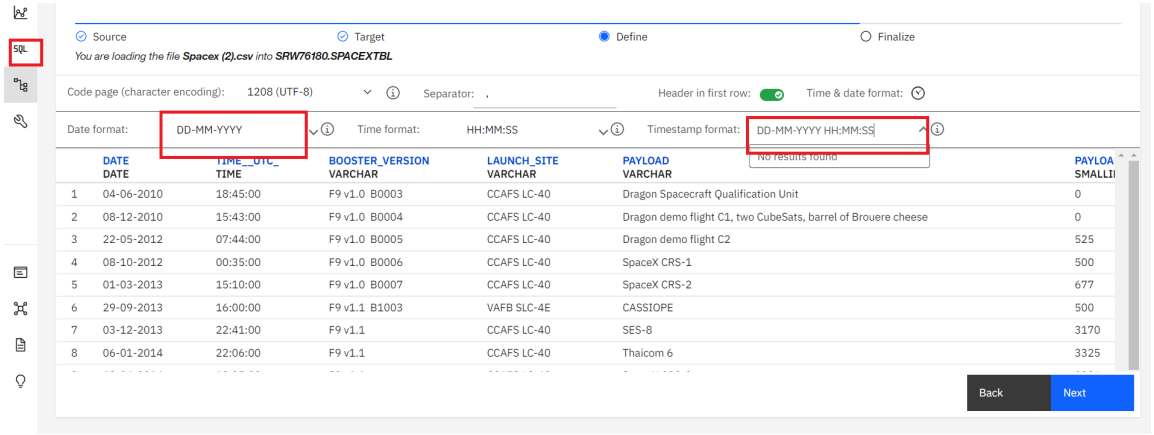


Changes to be considered when having DB2 instance with the new UI having Go to UI screen

- Refer to this insruction in this link for viewing the new 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.



In [1]:
```
!pip install sqlalchemy==1.3.9
!pip install ibm_db_sa
!pip install ipython-sql
```

```
Collecting sqlalchemy==1.3.9
  Downloading SQLAlchemy-1.3.9.tar.gz (6.0 MB)
     |████████████████████████████████| 6.0 MB 14.5 MB/s eta 0:00:01
Building wheels for collected packages: sqlalchemy
  Building wheel for sqlalchemy (setup.py) ... done
  Created wheel for sqlalchemy: filename=SQLAlchemy-1.3.9-cp38-cp38-linux_x86_64.whl
size=1209506 sha256=8fcc47c0e919a9f01b0c21e23c63b72c6cb9554d48d4fd991e2a4bd3b799a182
  Stored in directory: /tmp/wsuser/.cache/pip/wheels/cb/43/46/fa638f2422554332b7865d
600275b24568bf60e76104a94bb4
Successfully built sqlalchemy
Installing collected packages: sqlalchemy
  Attempting uninstall: sqlalchemy
    Found existing installation: SQLAlchemy 1.4.22
    Uninstalling SQLAlchemy-1.4.22:
      Successfully uninstalled SQLAlchemy-1.4.22
Successfully installed sqlalchemy-1.3.9
Requirement already satisfied: ibm_db_sa in /opt/conda/envs/Python-3.8-main/lib/pyth
on3.8/site-packages (0.3.7)
Requirement already satisfied: ibm-db>=2.0.0 in /opt/conda/envs/Python-3.8-main/lib/
python3.8/site-packages (from ibm_db_sa) (3.0.4)
Requirement already satisfied: sqlalchemy>=0.7.3 in /opt/conda/envs/Python-3.8-main/
lib/python3.8/site-packages (from ibm_db_sa) (1.3.9)
Collecting ipython-sql
  Downloading ipython_sql-0.4.0-py3-none-any.whl (19 kB)
Requirement already satisfied: six in /opt/conda/envs/Python-3.8-main/lib/python3.8/
site-packages (from ipython-sql) (1.15.0)
Requirement already satisfied: sqlalchemy>=0.6.7 in /opt/conda/envs/Python-3.8-main/
lib/python3.8/site-packages (from ipython-sql) (1.3.9)
Requirement already satisfied: ipython-genutils>=0.1.0 in /opt/conda/envs/Python-3.8
-main/lib/python3.8/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: ipython>=1.0 in /opt/conda/envs/Python-3.8-main/lib/p
ython3.8/site-packages (from ipython-sql) (7.27.0)
Collecting prettytable<1
  Downloading prettytable-0.7.2.zip (28 kB)
Collecting sqlparse
  Downloading sqlparse-0.4.2-py3-none-any.whl (42 kB)
     |████████████████████████████████| 42 kB 2.6 MB/s  eta 0:00:01
Requirement already satisfied: setuptools>=18.5 in /opt/conda/envs/Python-3.8-main/l
ib/python3.8/site-packages (from ipython>=1.0->ipython-sql) (52.0.0.post20211006)
Requirement already satisfied: traitlets>=4.2 in /opt/conda/envs/Python-3.8-main/li
b/python3.8/site-packages (from ipython>=1.0->ipython-sql) (5.0.5)
Requirement already satisfied: backcall in /opt/conda/envs/Python-3.8-main/lib/pytho
n3.8/site-packages (from ipython>=1.0->ipython-sql) (0.2.0)
Requirement already satisfied: decorator in /opt/conda/envs/Python-3.8-main/lib/pyth
on3.8/site-packages (from ipython>=1.0->ipython-sql) (5.0.9)
Requirement already satisfied: pickleshare in /opt/conda/envs/Python-3.8-main/lib/py
thon3.8/site-packages (from ipython>=1.0->ipython-sql) (0.7.5)
Requirement already satisfied: pexpect>4.3 in /opt/conda/envs/Python-3.8-main/lib/py
thon3.8/site-packages (from ipython>=1.0->ipython-sql) (4.8.0)
Requirement already satisfied: jedi>=0.16 in /opt/conda/envs/Python-3.8-main/lib/pyt
hon3.8/site-packages (from ipython>=1.0->ipython-sql) (0.17.2)
Requirement already satisfied: matplotlib-inline in /opt/conda/envs/Python-3.8-main/
lib/python3.8/site-packages (from ipython>=1.0->ipython-sql) (0.1.2)
Requirement already satisfied: pygments in /opt/conda/envs/Python-3.8-main/lib/pytho
n3.8/site-packages (from ipython>=1.0->ipython-sql) (2.9.0)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /opt/
```

```
conda/envs/Python-3.8-main/lib/python3.8/site-packages (from ipython>=1.0->ipython-s
ql) (3.0.20)
Requirement already satisfied: parso<0.8.0,>=0.7.0 in /opt/conda/envs/Python-3.8-mai
n/lib/python3.8/site-packages (from jedi>=0.16->ipython>=1.0->ipython-sql) (0.7.0)
Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/envs/Python-3.8-main/li
b/python3.8/site-packages (from pexpect>4.3->ipython>=1.0->ipython-sql) (0.7.0)
Requirement already satisfied: wcwidth in /opt/conda/envs/Python-3.8-main/lib/python
3.8/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)
Building wheels for collected packages: prettytable
  Building wheel for prettytable (setup.py) ... done
  Created wheel for prettytable: filename=prettytable-0.7.2-py3-none-any.whl size=13
700 sha256=fa982d74cf33b1ed21936c1f2271c7b91cfd6702b8a806112626d46e8142994d
  Stored in directory: /tmp/wsuser/.cache/pip/wheels/48/6d/77/9517cb933af254f51a446f
1a5ec9c2be3e45f17384940bce68
Successfully built prettytable
Installing collected packages: sqlparse, prettytable, ipython-sql
Successfully installed ipython-sql-0.4.0 prettytable-0.7.2 sqlparse-0.4.2
```

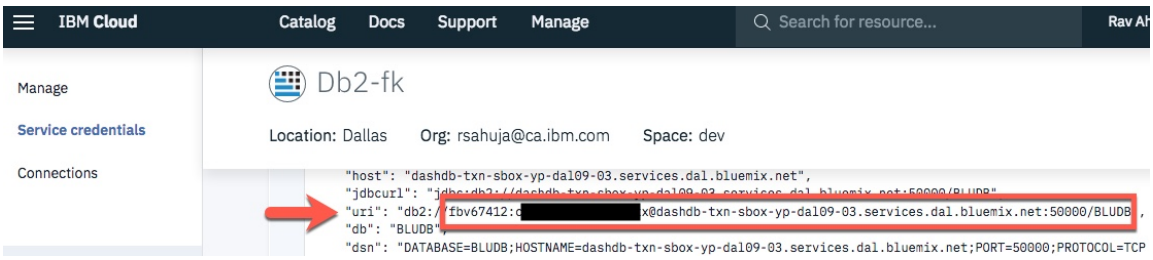
Connect to the database

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

In [2]:
```
%load_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.

- 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 [4]:
```
%sql ibm_db_sa://jjk92789:mUgin2bu22IbDNHE@125f9f61-9715-46f9-9399-c8177b21803b.c1o
```

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 [9]:
```
%%sql

select distinct Launch_Site from spacextbl
```

```
 * ibm_db_sa://jjk92789:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde0
0.databases.appdomain.cloud:30426/bludb
Done.
```

Out[9]:

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 [10]:
```
%%sql

select * from spacextbl where Launch_Site LIKE 'CCA%' limit 5;
```

```
 * ibm_db_sa://jjk92789:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde0
0.databases.appdomain.cloud:30426/bludb
Done.
```

Out[10]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	cust
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	S
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	(
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	(
2012-08-10	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	

Task 3

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

In [11]:
```
%%sql

select sum(PAYLOAD_MASS__KG_) from spacextbl where Customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://jjk92789:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde0
0.databases.appdomain.cloud:30426/bludb
Done.
```

Out[11]:

1
45596

Task 4

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

In [15]:
```
%%sql

select avg(PAYLOAD_MASS__KG_) from spacextbl where Booster_Version LIKE 'F9 v1.1';
```

```
 * ibm_db_sa://jjk92789:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde0
0.databases.appdomain.cloud:30426/bludb
Done.
```

Out[15]:

1
2928

Task 5

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

Hint:Use min function

In [27]:
```
%%sql

select min(Date) as min_date from spacextbl where Landing__Outcome = 'Success (grou
```

```
 * ibm_db_sa://jjk92789:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde0
0.databases.appdomain.cloud:30426/bludb
Done.
```

Out[27]:

min_date
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 [29]:
```
%%sql

select Booster_Version from spacextbl where (PAYLOAD_MASS__KG_> 4000 and PAYLOAD_MA
and (Landing__Outcome = 'Success (drone ship)');
```

```
 * ibm_db_sa://jjk92789:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde0
0.databases.appdomain.cloud:30426/bludb
Done.
```

Out[29]:

booster_version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Task 7

List the total number of successful and failure mission outcomes

In [6]:
```
%%sql

select Mission_Outcome, count(Mission_Outcome) as counts from spacextbl group by Mi
```

```
 * ibm_db_sa://jjk92789:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde0
0.databases.appdomain.cloud:30426/bludb
Done.
```

Out[6]:

mission_outcome	counts
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Task 8

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

In [32]:
```
%%sql

select Booster_Version, PAYLOAD_MASS__KG_ from spacextbl where PAYLOAD_MASS__KG_ =
```

```
 * ibm_db_sa://jjk92789:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde0
0.databases.appdomain.cloud:30426/bludb
Done.
```

Out[32]:

booster_version	payload_mass__kg_
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 [37]:
```
%%sql

select Landing__Outcome, Booster_Version, Launch_Site from spacextbl where Landing_
```

```
 * ibm_db_sa://jjk92789:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde0
0.databases.appdomain.cloud:30426/bludb
Done.
```

Out[37]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
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 [39]:
```
%%sql

select Landing__Outcome, count(*) as LandingCounts from spacextbl where Date betwee
group by Landing__Outcome
order by count(*) desc;
```

```
 * ibm_db_sa://jjk92789:***@125f9f61-9715-46f9-9399-c8177b21803b.c1ogj3sd0tgtu0lqde0
0.databases.appdomain.cloud:30426/bludb
Done.
```

Out[39]:

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