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

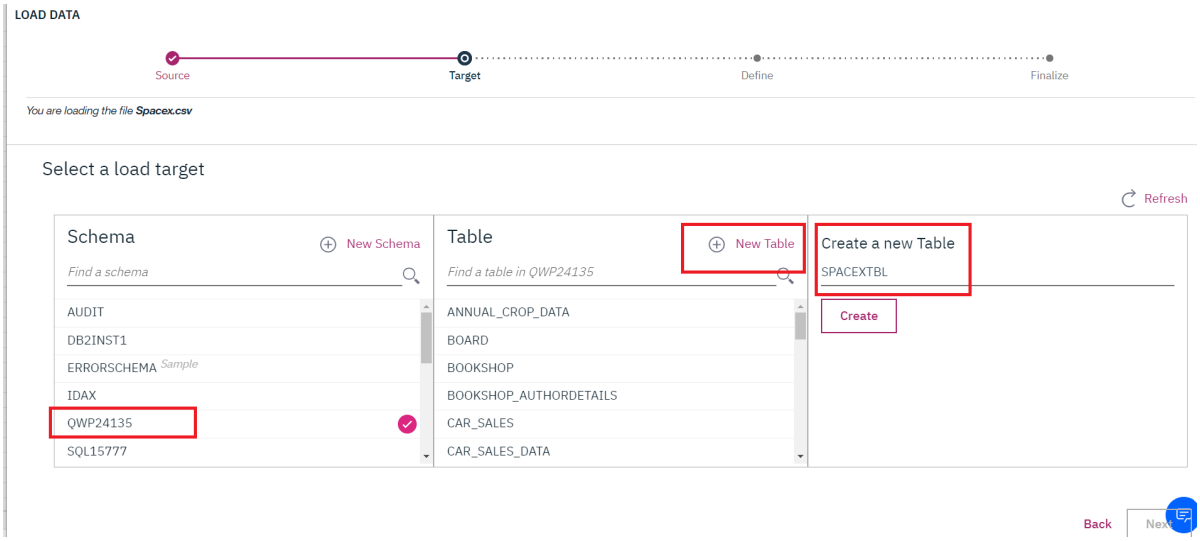
This assignment requires you to load the spacex dataset.

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

Store the dataset in database table

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



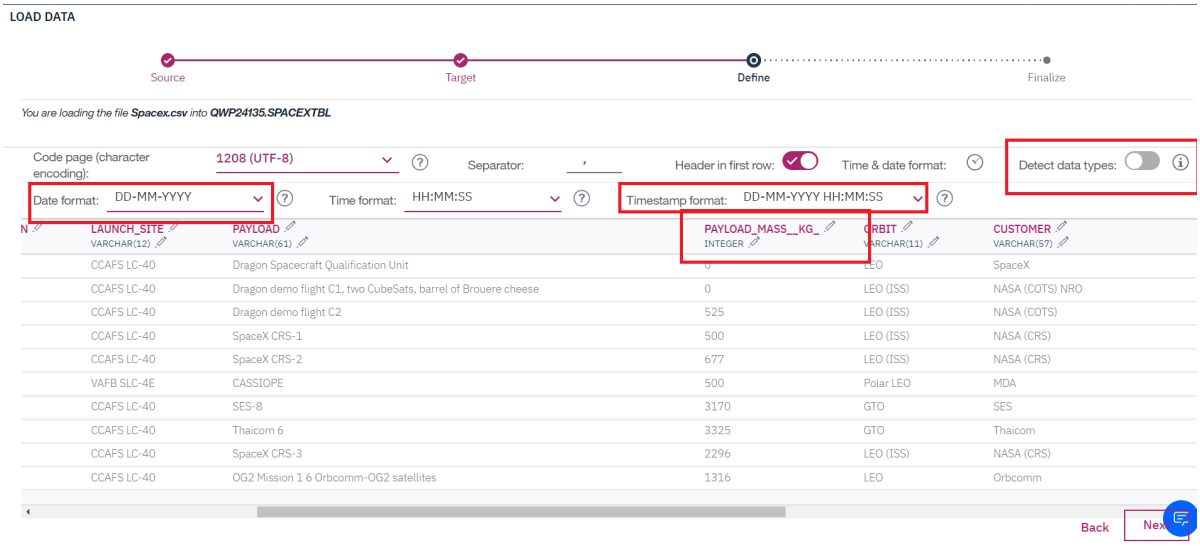
Now open the Db2 console, open the LOAD tool, Select / Drag the .CSV file for the dataset, Next create a New Table, and then follow the steps on-screen instructions to load the data. Name the new table as follows:

SPACEXDATASET

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

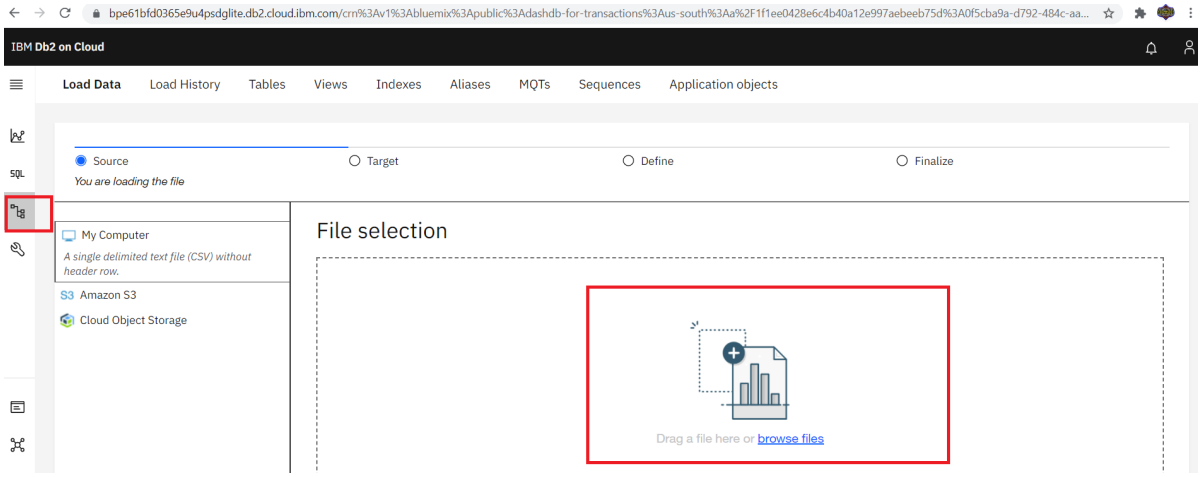
Note:While loading Spacex dataset, ensure that detect datatypes is disabled. Later click on the pencil icon(edit option).

1. Change the Date Format by manually typing DD-MM-YYYY and timestamp format as DD-MM-YYYY HH\:MM:SS
2. Change the PAYLOAD*MASS*_KG_ datatype to INTEGER.

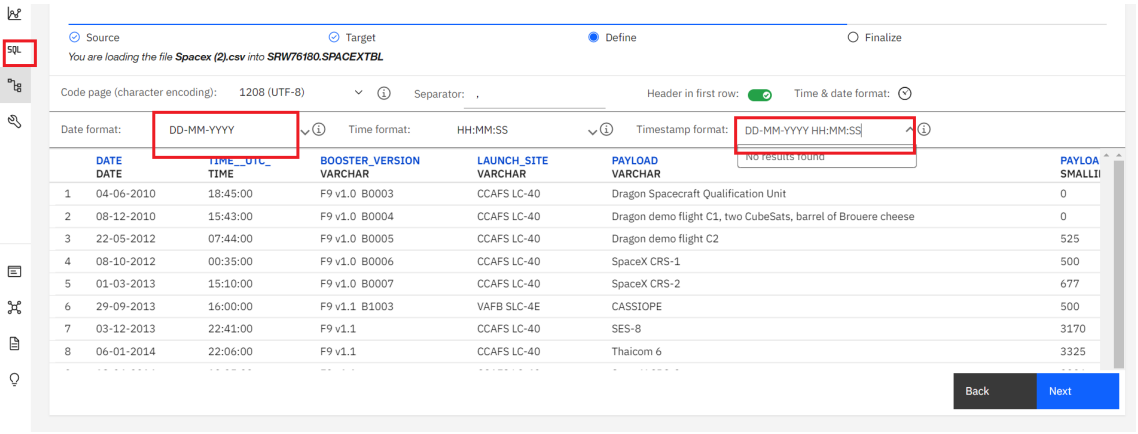


Changes to be considered when having DB2 instance with the new UI having Go to UI screen

- 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
```

```
Requirement already satisfied: sqlalchemy==1.3.9 in c:\users\deept\anaconda_files\li
b\site-packages (1.3.9)
```

Connect to the database

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

In [2]:
```
%load_ext sql
```

In [3]:
```
import csv, sqlite3

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

In [4]:
```
!pip install -q pandas==1.1.5
```

In [5]:
```
%sql sqlite:///my_data1.db
```

In [6]:
```
import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")
```

```
C:\Users\deept\Anaconda_files\lib\site-packages\pandas\core\generic.py:2605: UserWar
ning: The spaces in these column names will not be changed. In pandas versions < 0.1
4, spaces were converted to underscores.
  sql.to_sql(
```

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 [7]:
```
%%sql
SELECT DISTINCT Launch_Site
FROM SPACEXTBL;
```

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

Out[7]:

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

Task 2

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

In [8]:
```
%%sql

SELECT *
FROM SPACEXTBL
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5;
```

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

Out[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	I
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	
08-12-2010	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	

22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2		525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1		500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2		677	LEO (ISS)	NASA (CRS)

Task 3

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

In [9]:
```
%%sql

SELECT SUM(PAYLOAD_MASS__KG_    ) AS Total_Payload_Mass
FROM SPACEXTBL
GROUP BY Customer
HAVING Customer = 'NASA (CRS)';
```

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

Out[9]:

Total_Payload_Mass
45596

Task 4

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

In [10]:
```
%%sql

SELECT Booster_Version, AVG(PAYLOAD_MASS__KG_   ) AS Average_Payload_Mass
FROM SPACEXTBL
GROUP BY Booster_Version
HAVING Booster_Version LIKE 'F9 v1.1' ;
```

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

Out[10]:

Booster_Version	Average_Payload_Mass
F9 v1.1	2928.4

Task 5

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

Hint:Use min function

In [11]:
```
%%sql

SELECT "Landing _Outcome", min(Date) as first_succesful_landing
FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE 'Success (ground pad)';
```

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

Out[11]:

Landing _Outcome	first_succesful_landing
Success (ground pad)	01-05-2017

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 [12]:

```
%%sql

SELECT Booster_Version, min(Date)
FROM SPACEXTBL
WHERE ("Landing _Outcome" LIKE 'Success (drone ship)') AND (PAYLOAD_MASS__KG_    ) >
```

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

Out[12]:

Booster_Version	min(Date)
F9 FT B1022	06-05-2016

Task 7

List the total number of successful and failure mission outcomes

In [13]:

```
%%sql

SELECT CASE WHEN Mission_Outcome LIKE 'Success%' THEN 'Success'
            WHEN Mission_Outcome LIKE 'Failure%' THEN 'Failure'
            END AS Outcome, COUNT(*)
FROM SPACEXTBL
GROUP BY Outcome;
```

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

Out[13]:

Outcome	COUNT(*)
Failure	1
Success	100

Task 8

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

In [14]:

```
%%sql

SELECT Booster_Version
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_  = (SELECT MAX(PAYLOAD_MASS__KG_        ) FROM SPACEXTBL);
```

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

Out[14]:

Booster_Version
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

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.

In [15]:
```
%%sql

WITH sub as(SELECT substr(Date, 4, 2) AS MONTH, "Landing _Outcome", Booster_Version,
FROM SPACEXTBL
WHERE ("Landing _Outcome" LIKE 'Failure (drone ship)') AND substr(Date,7,4)='2015')


SELECT CASE WHEN MONTH = '01' THEN 'January'
            WHEN MONTH = '02' THEN 'February'
            WHEN MONTH = '03' THEN 'March'
            WHEN MONTH = '04' THEN 'April'
            WHEN MONTH = '05' THEN 'May'
            WHEN MONTH = '06' THEN 'June'
            WHEN MONTH = '07' THEN 'July'
            WHEN MONTH = '08' THEN 'August'
            WHEN MONTH = '09' THEN 'September'
            WHEN MONTH = '10' THEN 'October'
            WHEN MONTH = '11' THEN 'November'
            WHEN MONTH = '12' THEN 'December'
            END as Month_name, "Landing _Outcome", Booster_Version, Launch_Site
    FROM sub;
```

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

Out[15]:

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 successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

In [16]:
```
%%sql

SELECT Count(*) as sucessful_landings
FROM SPACEXTBL
WHERE ("Landing _Outcome" LIKE 'Success%') AND (Date BETWEEN '04-06-2010' AND '20-03
ORDER BY Date DESC;
```

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

Out[16]:

sucessful_landings
34

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