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

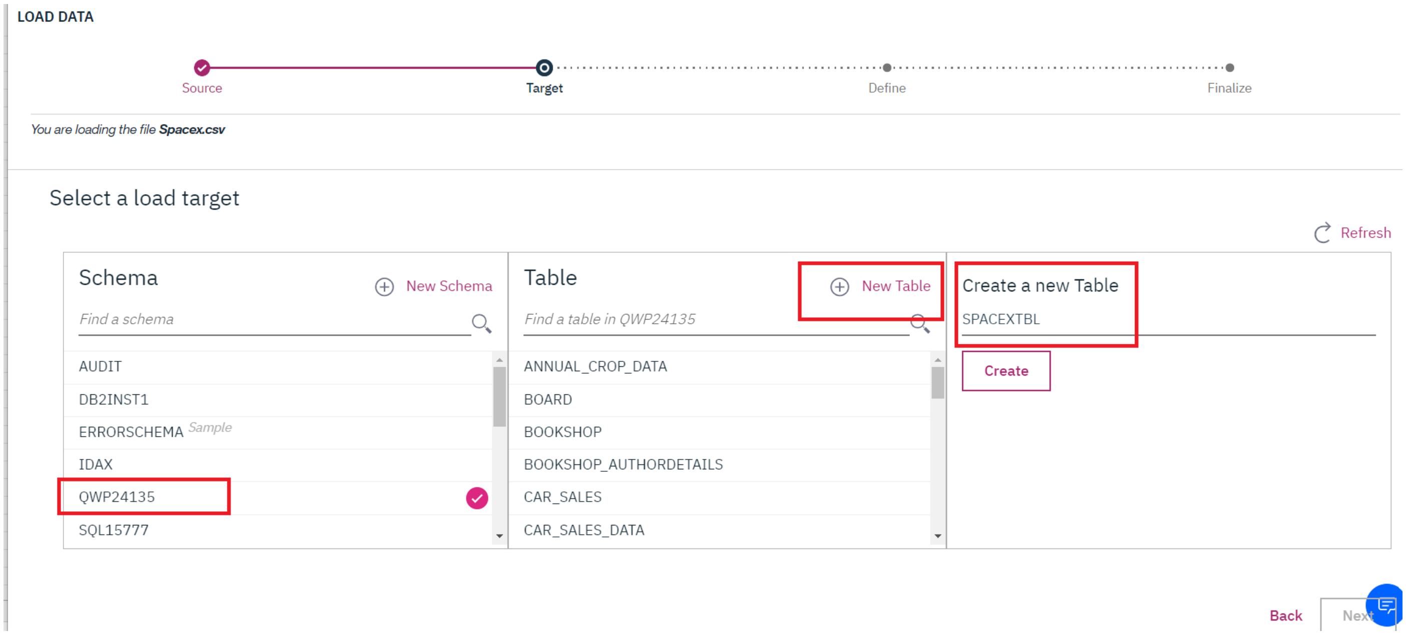
This assignment requires you to load the spacex dataset.

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 (https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01)

Store the dataset in database table

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



You are loading the file ***Spacex.csv***

Select a load target

Create a new Table: SPACEXTBL

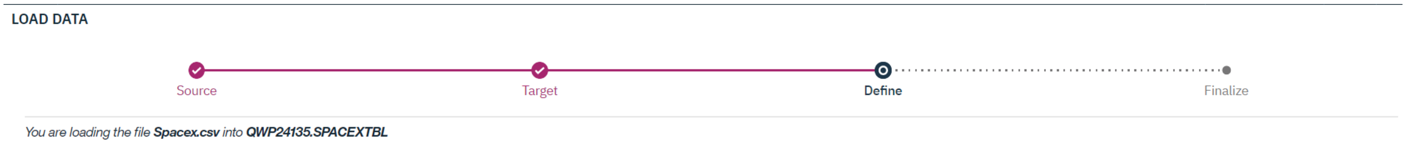
Now open the Db2 console, open the LOAD tool, Select / Drag the .CSV file for the dataset, Next create a New Table, and then follow the steps on-screen instructions to load the data. Name the new table as follows:

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.

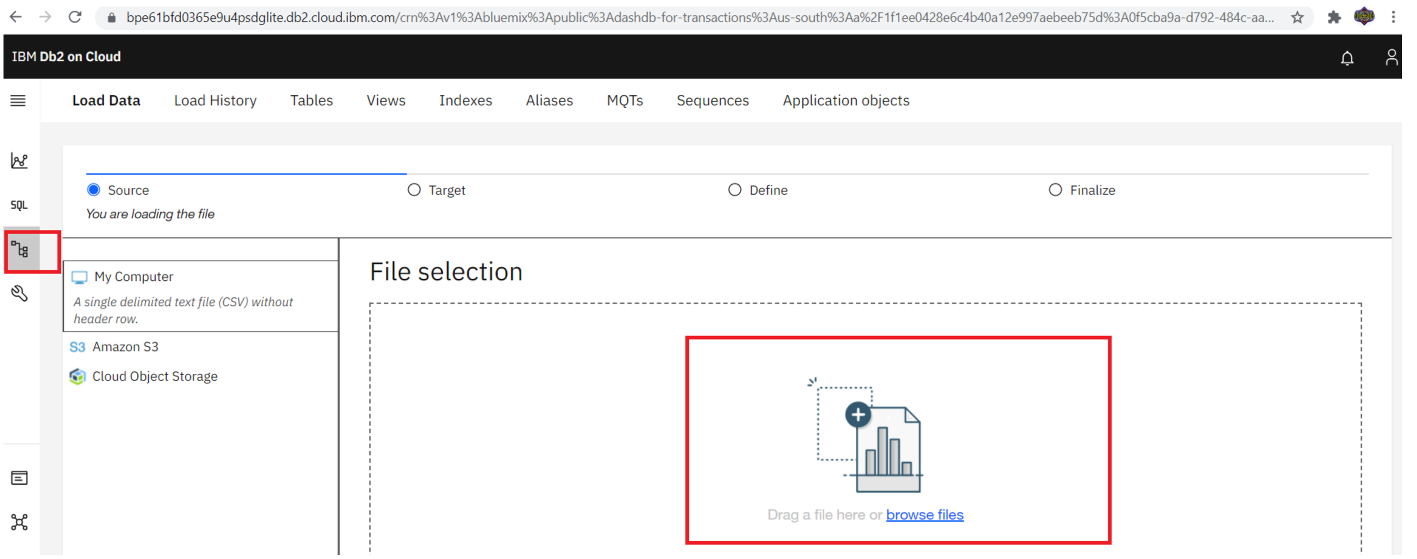


LOAD DATA

*You are loading the file **Spacex.csv** into **QWP24135.SPACEXTBL***

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

- Refer to this insruction in this link (https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Labs_Coursera_V5/labs/Lab%20-%20Sign%20up%20for%20IBM%20Cloud%20-%20Create%20Db2%20service%20instance%20-%20Get%20started%20with%20the%20Db2%20console/instructional-labs.md.html?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01) 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.



bpe61bfd0365e9u4psdglite.db2.cloud.ibm.com/crn%3Av1%3Abluemix%3Apublic%3Adashdb-for-transactions%3Aus-south%3Aa%2F1f1ee0428e6c4b40a12e997aebeeb75d%3A0f5cba9a-d792-484c-aa...
IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
SQL
Source
Target
Define
Finalize

You are loading the file

File selection

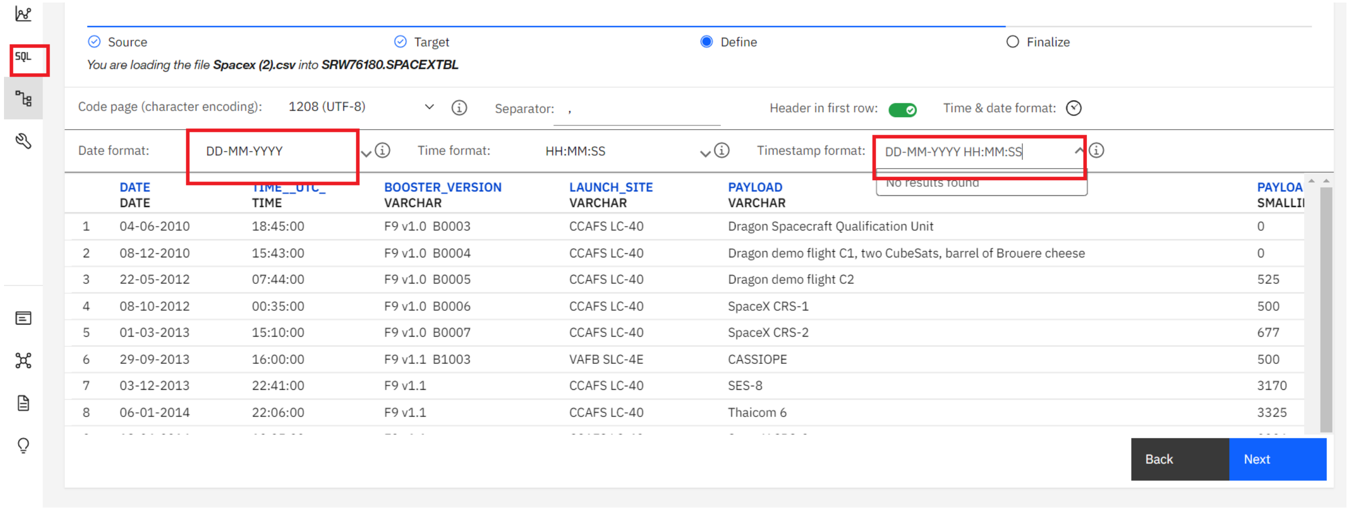
My Computer

A single delimited text file (CSV) without header row.

S3 Amazon S3
Cloud Object Storage

Drag a file here or browse files

- Once done select the schema andload the file.



*You are loading the file **Spacex (2).csv** into **SRW76180.SPACEXTBL***

Code page (character encoding): 1208 (UTF-8)
Separator: ,
Header in first row: ●
Time & date format:
Date format: DD-MM-YYYY
Time format: HH:MM:SS
Timestamp format: DD-MM-YYYY HH:MM:SS
No results found

	DATE DATE	TIME__UTC_ TIME	BOOSTER_VERSION VARCHAR	LAUNCH_SITE VARCHAR	PAYLOAD VARCHAR	PAYLOA SMALLI
1	04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0
2	08-12-2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0
3	22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525
4	08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500
5	01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677
6	29-09-2013	16:00:00	F9 v1.1 B1003	VAFB SLC-4E	CASSIOPE	500
7	03-12-2013	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170
8	06-01-2014	22:06:00	F9 v1.1	CCAFS LC-40	Thaicom 6	3325

In [1]:

```
!pip install sqlalchemy==1.3.9
!pip install ibm_db_sa
!pip install ipython-sql
```

```
Requirement already satisfied: sqlalchemy==1.3.9 in /home/jupyterlab/conda/en
vs/python/lib/python3.6/site-packages (1.3.9)
Requirement already satisfied: ibm_db_sa in /home/jupyterlab/conda/envs/pytho
n/lib/python3.6/site-packages (0.3.3)
Requirement already satisfied: sqlalchemy>=0.7.3 in /home/jupyterlab/conda/en
vs/python/lib/python3.6/site-packages (from ibm_db_sa) (1.3.9)
Requirement already satisfied: ipython-sql in /home/jupyterlab/conda/envs/pyt
hon/lib/python3.6/site-packages (0.3.9)
Requirement already satisfied: ipython>=1.0 in /home/jupyterlab/conda/envs/py
thon/lib/python3.6/site-packages (from ipython-sql) (7.16.1)
Requirement already satisfied: sqlparse in /home/jupyterlab/conda/envs/pytho
n/lib/python3.6/site-packages (from ipython-sql) (0.4.1)
Requirement already satisfied: prettytable in /home/jupyterlab/conda/envs/pyt
hon/lib/python3.6/site-packages (from ipython-sql) (2.1.0)
Requirement already satisfied: ipython-genutils>=0.1.0 in /home/jupyterlab/co
nda/envs/python/lib/python3.6/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: sqlalchemy>=0.6.7 in /home/jupyterlab/conda/en
vs/python/lib/python3.6/site-packages (from ipython-sql) (1.3.9)
Requirement already satisfied: six in /home/jupyterlab/conda/envs/python/lib/
python3.6/site-packages (from ipython-sql) (1.15.0)
Requirement already satisfied: decorator in /home/jupyterlab/conda/envs/pytho
n/lib/python3.6/site-packages (from ipython>=1.0->ipython-sql) (4.4.2)
Requirement already satisfied: backcall in /home/jupyterlab/conda/envs/pytho
n/lib/python3.6/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 i
n /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipytho
n>=1.0->ipython-sql) (3.0.19)
Requirement already satisfied: pexpect; sys_platform != "win32" in /home/jupy
terlab/conda/envs/python/lib/python3.6/site-packages (from ipython>=1.0->ipyt
hon-sql) (4.8.0)
Requirement already satisfied: pygments in /home/jupyterlab/conda/envs/pytho
n/lib/python3.6/site-packages (from ipython>=1.0->ipython-sql) (2.9.0)
Requirement already satisfied: traitlets>=4.2 in /home/jupyterlab/conda/envs/
python/lib/python3.6/site-packages (from ipython>=1.0->ipython-sql) (4.3.3)
Requirement already satisfied: jedi>=0.10 in /home/jupyterlab/conda/envs/pyth
on/lib/python3.6/site-packages (from ipython>=1.0->ipython-sql) (0.17.2)
Requirement already satisfied: pickleshare in /home/jupyterlab/conda/envs/pyt
hon/lib/python3.6/site-packages (from ipython>=1.0->ipython-sql) (0.7.5)
Requirement already satisfied: setuptools>=18.5 in /home/jupyterlab/conda/env
s/python/lib/python3.6/site-packages (from ipython>=1.0->ipython-sql) (49.6.
0.post20210108)
Requirement already satisfied: wcwidth in /home/jupyterlab/conda/envs/python/
lib/python3.6/site-packages (from prettytable->ipython-sql) (0.2.5)
Requirement already satisfied: importlib-metadata; python_version < "3.8" in
/home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from prettyta
ble->ipython-sql) (4.6.1)
Requirement already satisfied: ptyprocess>=0.5 in /home/jupyterlab/conda/env
s/python/lib/python3.6/site-packages (from pexpect; sys_platform != "win32"->
ipython>=1.0->ipython-sql) (0.7.0)
Requirement already satisfied: parso<0.8.0,>=0.7.0 in /home/jupyterlab/conda/
envs/python/lib/python3.6/site-packages (from jedi>=0.10->ipython>=1.0->ipyth
on-sql) (0.7.1)
Requirement already satisfied: zipp>=0.5 in /home/jupyterlab/conda/envs/pytho
n/lib/python3.6/site-packages (from importlib-metadata; python_version < "3.
8"->prettytable->ipython-sql) (3.5.0)
Requirement already satisfied: typing-extensions>=3.6.4; python_version < "3.
8" in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from im
portlib-metadata; python_version < "3.8"->prettytable->ipython-sql) (3.10.0.
0)
```

Connect to the database

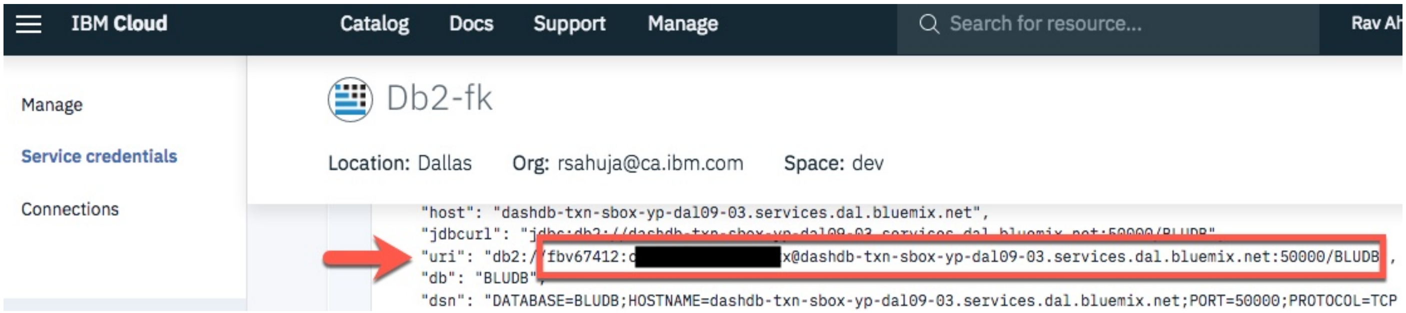
Let us first load the SQL extension and establish a connection with the database

In [3]:

```
%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://



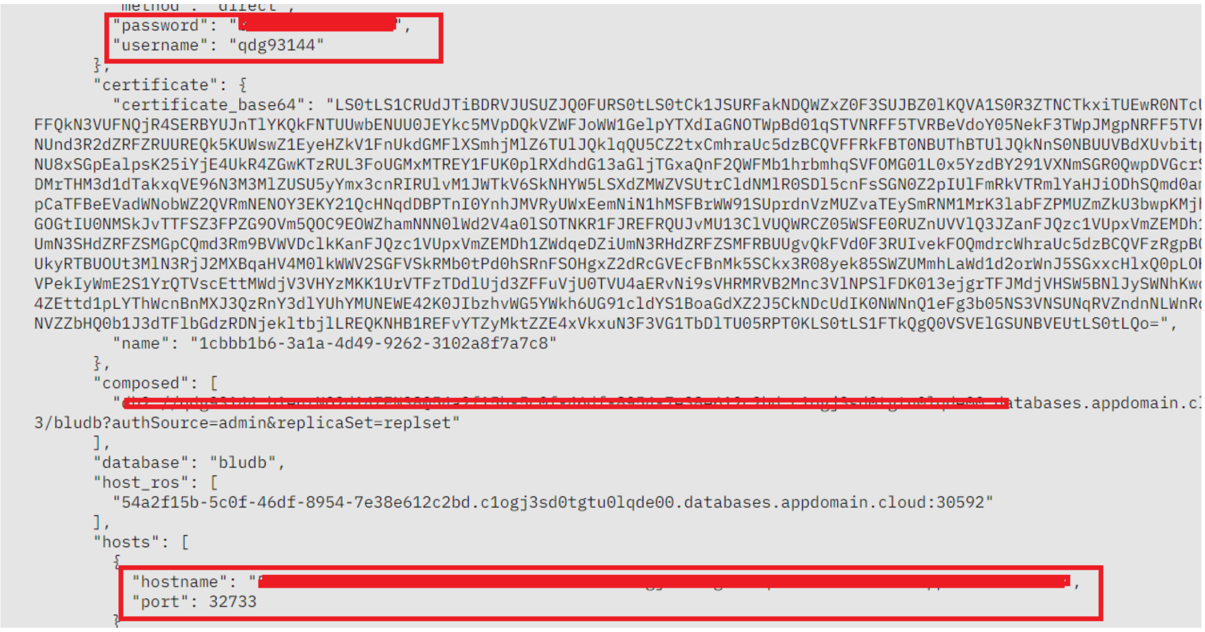
Location: Dallas Org: rsahuja@ca.ibm.com Space: dev

```
"host": "dashdb-txn-sbox-yp-dal09-03.services.dal.bluemix.net",
"jdbcurl": "jdbc:db2://dashdb-txn-sbox-yp-dal09-03.services.dal.bluemix.net:50000/BLUDB",
"uri": "db2://fbv67412:c[redacted]x@dashdb-txn-sbox-yp-dal09-03.services.dal.bluemix.net:50000/BLUDB",
"db": "BLUDB",
"dsn": "DATABASE=BLUDB;HOSTNAME=dashdb-txn-sbox-yp-dal09-03.services.dal.bluemix.net;PORT=50000;PROTOCOL=TCP
```

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.



```
            "method": "direct",
            "password": "[redacted]",
            "username": "qdg93144"
        },
        "certificate": {
            "certificate_base64": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURFakNDQWZxZ0F3SUJBZ0lKQVA1S0R3ZTNCTkxiTUEwR0NTc
   FFQkN3VUFNQjR4SERBYUJnTlYKQkFNTUUwbENUU0JEYkc5MVpDQkVZWFJoWW1GelpYTXdIaGNOTWpBd01qSTVNRFF5TVRBeVdoY05NekF3TWpJMgpNRFF5TV
   NUnd3R2dZRFZRUUREQk5KUWswZ1EyeHZkV1FnUkdGMFlXSmhjMlZ6TUlJQklqQU5CZ2txCmhraUc5dzBCQVFFRkFBT0NBUThBTUlJQkNnS0NBUUVBdXUvbit
   NU8xSGpEalpsK25iYjE4UkR4ZGwKTzRUL3FoUGMxMTREY1FUK0plRXdhdG13aGljTGxaQnF2QWFMb1hrbmhqSVFOMG01L0x5YzdBY291VXNmSGR0QwpDVGcr
   DMrTHM3d1dTakxqVE96N3M3MlZUSU5yYmx3cnRIRUlvM1JWTkV6SkNHYW5LSXdZMWZVSUtrCldNMlR0SDl5cnFsSGN0Z2pIUlFmRkVTRmlYaHJiODhSQmd0a
   pCaTFBeEVadWNobWZ2QVRmNENOY3EKY21QcHNqdDBPTnI0YnhJMVRyUWxEemNiN1hMSFBrWW91SUprdnVzMUZvaTEySmRNM1MrK3labFZPMUZmZkU3bwpKMj
   GOGtIU0NMSkJvTTFSZ3FPZG9OVm5QOC9EOWZhamNNN0lWd2V4a0lSOTNKR1FJREFRQUJvMU13ClVUQWRCZ05WSFE0RUZnUVVlQ3JZanFJQzc1VUpxVmZEMDh
   UmN3SHdZRFZSMGpCQmd3Rm9BVWVDclkKanFJQzc1VUpxVmZEMDh1ZWdqeDZiUmN3RHdZRFZSMFRBUUgvQkFVd0F3RUIvekFOQmdrcWhraUc5dzBCQVFzRgpB
   UkyRTBUOUt3MlN3RjJ2MXBqaHV4M0lkWWV2SGFVSkRMb0tPd0hSRnFSOHgxZ2dRcGVEcFBnMk5SCkx3R08yek85SWZUMmhLaWd1d2orWnJ5SGxxcHlxQ0pLO
   VPekIyWmE2S1YrQTVscEttMWdjV3VHYzMKK1UrVTFzTDdlUjd3ZFFuVjU0TVU4aERvNi9sVHRMRVB2Mnc3VlNPSlFDK013ejgrTFJMdjVHSW5BNlJySWNhKw
   4ZEttd1pLYThWcnBnMXJ3QzRnY3dlYUhYMUNEWE42K0JIbzhvWG5YWkh6UG91cldYS1BoaGdXZ2J5CkNDcUdIK0NWNnQ1eFg3b05NS3VNSUNqRVZndnNLWnR
   NVZZbHQ0b1J3dTFlbGdzRDNjekltbjlLREQKNHB1REFvYTZyMktZZE4xVkxuN3F3VG1TbDlTU05RPT0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=",
            "name": "1cbbb1b6-3a1a-4d49-9262-3102a8f7a7c8"
        },
        "composed": [
            "[redacted]atabases.appdomain.cl
   3/bludb?authSource=admin&replicaSet=replset"
        ],
        "database": "bludb",
        "host_ros": [
            "54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:30592"
        ],
        "hosts": [
            {
              "hostname": "[redacted]",
              "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 [ ]:

In [ ]:

In [4]:

```
%sql ibm_db_sa://mns53034:9xk4cn2%5E2rxvr9ct@dashdb-txn-sbox-yp-lon02-04.services.eu-gb.bl
uemix.net:50000/BLUDB
```

Out[4]:

```
'Connected: mns53034@BLUDB'
```

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 [31]:

```
%sql select distinct LAUNCH_SITE from SPACEXDATASET
```

```
 * ibm_db_sa://mns53034:***@dashdb-txn-sbox-yp-lon02-04.services.eu-gb.bluemi
x.net:50000/BLUDB
Done.
```

Out[31]:

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 [32]:

```
%%sql
select * from SPACEXDATASET where LAUNCH_SITE LIKE 'CCA%'LIMIT 5;
```

```
 * ibm_db_sa://mns53034:***@dashdb-txn-sbox-yp-lon02-04.services.eu-gb.bluemi
x.net:50000/BLUDB
Done.
```

Out[32]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
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
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
2013-03-01	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 [13]:

```
%%sql

Select SUM(PAYLOAD_MASS__KG_) from SPACEXDATASET where Customer = 'NASA (CRS)';
```

```
 * ibm_db_sa://mns53034:***@dashdb-txn-sbox-yp-lon02-04.services.eu-gb.bluemi
x.net:50000/BLUDB
Done.
```

Out[13]:

1
45596

Task 4

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

In [14]:

```
%%sql

Select AVG(PAYLOAD_MASS__KG_) from SPACEXDATASET where BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://mns53034:***@dashdb-txn-sbox-yp-lon02-04.services.eu-gb.bluemi
x.net:50000/BLUDB
Done.
```

Out[14]:

1
2928.400000

Task 5

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

Hint:Use min function

In [17]:

```
%%sql

Select MIN(DATE) from SPACEXDATASET where LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://mns53034:***@dashdb-txn-sbox-yp-lon02-04.services.eu-gb.bluemi
x.net:50000/BLUDB
Done.
```

Out[17]:

1
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 [20]:

```
%%sql

Select distinct PAYLOAD from SPACEXDATASET where LANDING__OUTCOME = 'Success (drone ship)'
AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000);
```

```
 * ibm_db_sa://mns53034:***@dashdb-txn-sbox-yp-lon02-04.services.eu-gb.bluemi
x.net:50000/BLUDB
Done.
```

Out[20]:

payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Task 7

List the total number of successful and failure mission outcomes

In [22]:

```
%%sql

Select MISSION_OUTCOME, COUNT(MISSION_OUTCOME) from SPACEXDATASET
GROUP BY MISSION_OUTCOME ORDER BY COUNT(MISSION_OUTCOME) DESC ;
```

```
 * ibm_db_sa://mns53034:***@dashdb-txn-sbox-yp-lon02-04.services.eu-gb.bluemi
x.net:50000/BLUDB
Done.
```

Out[22]:

mission_outcome	2
Success	99
Failure (in flight)	1
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 [30]:

```
%%sql

Select distinct BOOSTER_VERSION from SPACEXDATASET where PAYLOAD_MASS__KG_ = (select MAX(P
AYLOAD_MASS__KG_) from SPACEXDATASET) ;
```

```
 * ibm_db_sa://mns53034:***@dashdb-txn-sbox-yp-lon02-04.services.eu-gb.bluemi
x.net:50000/BLUDB
Done.
```

Out[30]:

booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

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 BOOSTER_VERSION, LAUNCH_SITE from SPACEXDATASET where LANDING__OUTCOME = 'Failure
 (drone ship)' AND DATE LIKE '2015%'
```

```
 * ibm_db_sa://mns53034:***@dashdb-txn-sbox-yp-lon02-04.services.eu-gb.bluemi
x.net:50000/BLUDB
Done.
```

Out[37]:

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 [40]:

```
%%sql

Select LANDING__OUTCOME, COUNT(LANDING__OUTCOME) from SPACEXDATASET where (DATE between '2
010-06-04' AND '2017-03-20')
GROUP BY LANDING__OUTCOME ORDER BY COUNT(LANDING__OUTCOME) DESC ;
```

```
 * ibm_db_sa://mns53034:***@dashdb-txn-sbox-yp-lon02-04.services.eu-gb.bluemi
x.net:50000/BLUDB
Done.
```

Out[40]:

landing__outcome	2
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 (https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Labs_Coursera_V5/labs/Lab%20-%20String%20Patterns%20-%20Sorting%20-%20Grouping/instructional-labs.md.html?
  utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=
  SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01&origin=www.coursera.org)
- Hands-on Lab: Built-in functions (https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Labs_Coursera_V5/labs/Lab%20-%20Built-in%20functions%20/Hands-on_Lab__Built-in_Functions.md.html?
  utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=
  SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01&origin=www.coursera.org)
- Hands-on Lab : Sub-queries and Nested SELECT Statements (https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Labs_Coursera_V5/labs/Lab%20-%20Sub-queries%20and%20Nested%20SELECTs%20/instructional-labs.md.html?
  utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=
  SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01&origin=www.coursera.org)
- Hands-on Tutorial: Accessing Databases with SQL magic (https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Module%205/DB0201EN-Week3-1-3-SQLmagic.ipynb?
  utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=
  SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01)
- Hands-on Lab: Analyzing a real World Data Set (https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Module%205/DB0201EN-Week3-1-4-Analyzing.ipynb?
  utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=
  SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01)

Author(s)

Lakshmi Holla

Other Contributors

Rav Ahuja

Change log

Date	Version	Changed by	Change Description
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