cognitiveclass.ai logo

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

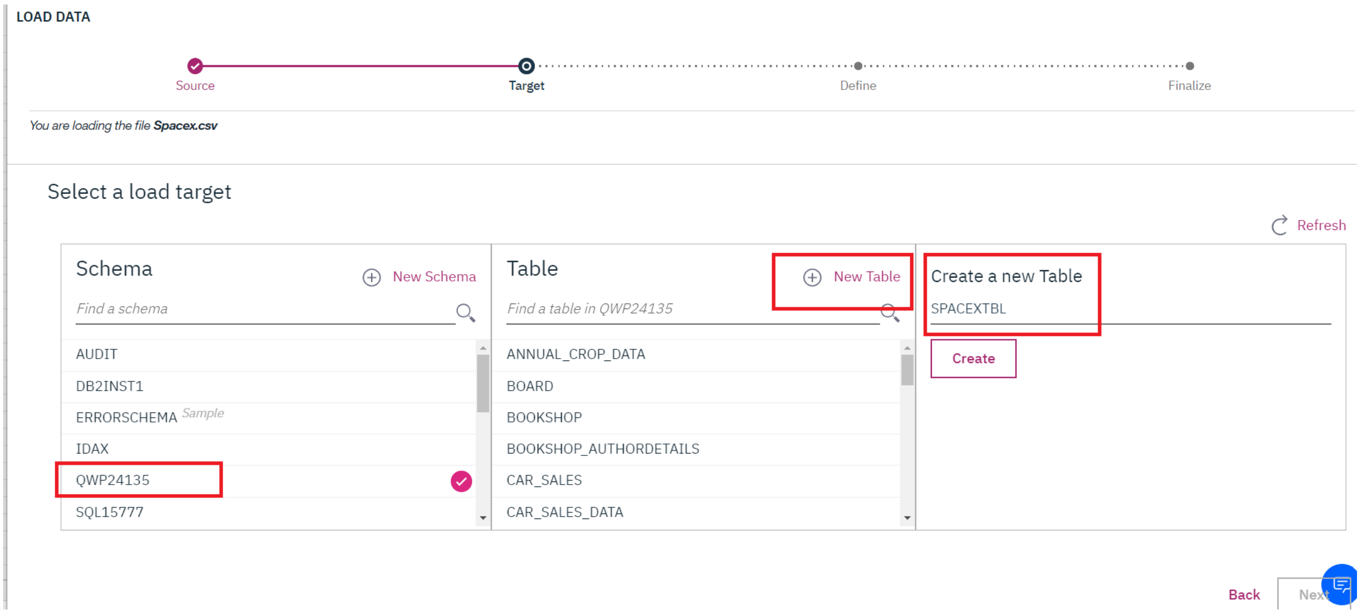
This assignment requires you to load the spacex dataset.

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
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.



LOAD DATA

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

Select a load target

Schema	Table	Create a new Table
AUDIT	ANNUAL_CROP_DATA	SPACEXTBL
DB2INST1	BOARD	
ERRORSCHEMA *Sample*	BOOKSHOP	
IDAX	BOOKSHOP_AUTHORDETAILS	
QWP24135 ✓	CAR_SALES	
SQL15777	CAR_SALES_DATA	

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:

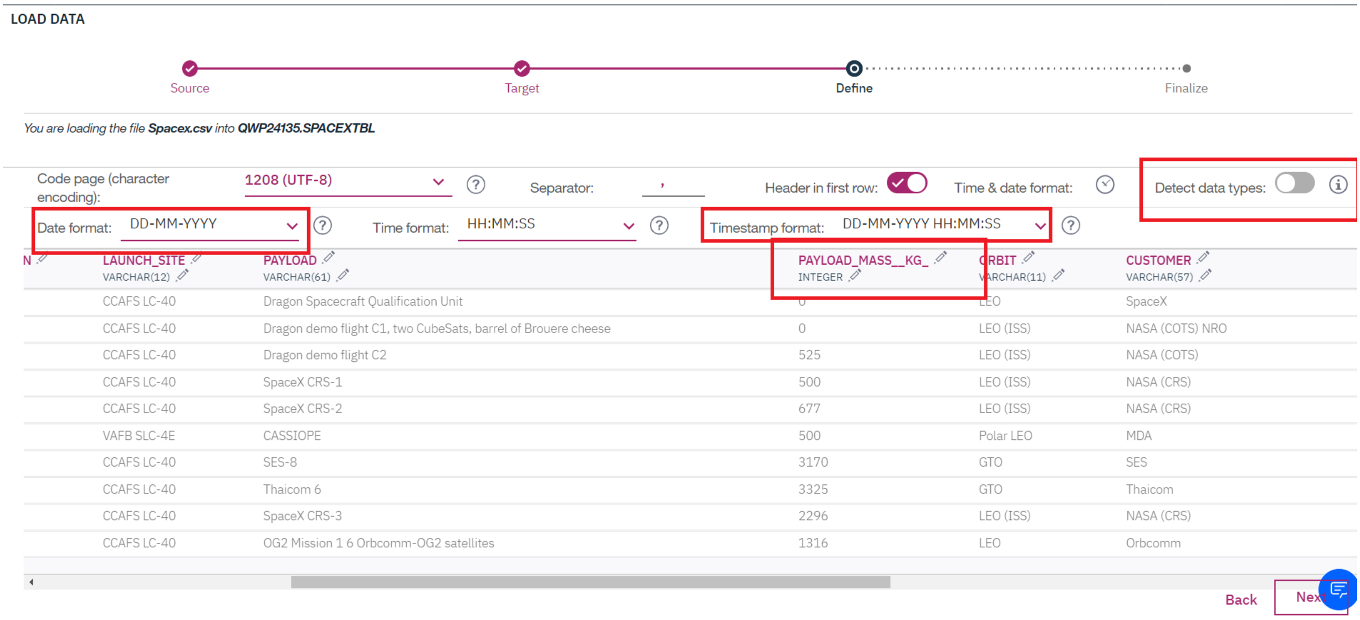
SPACEXDATASET

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.



LOAD DATA

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

Code page (character encoding): 1208 (UTF-8)

Separator: ,

Header in first row: ●

Time & date format:

Detect data types: ○

Date format: DD-MM-YYYY

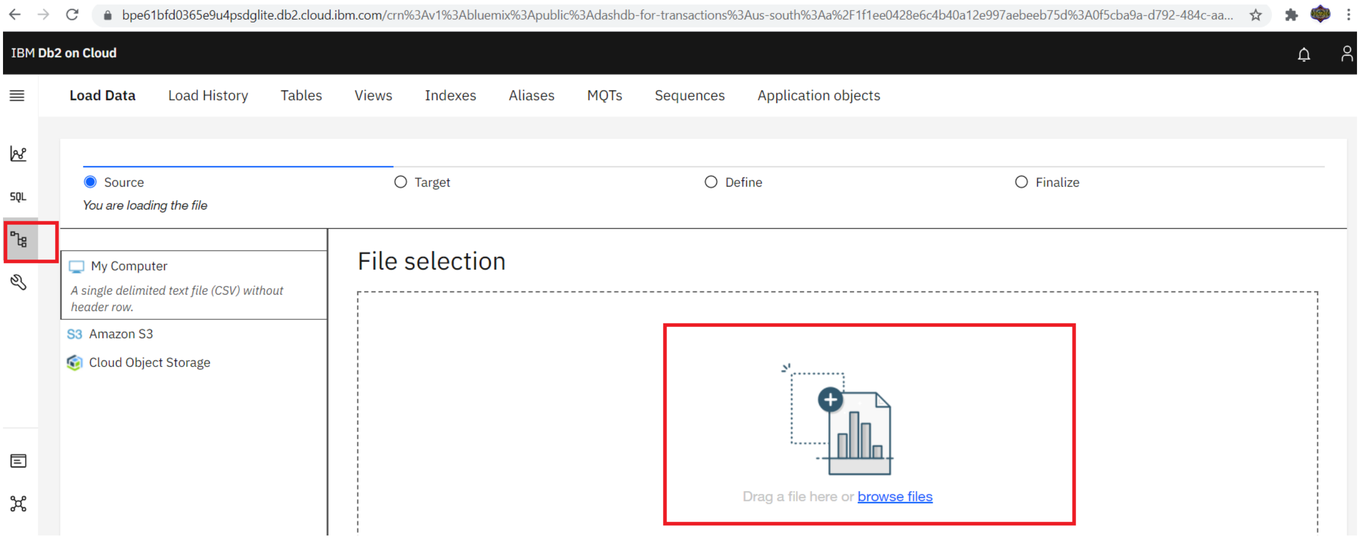
Time format: HH:MM:SS

Timestamp format: DD-MM-YYYY HH:MM:SS

LAUNCH_SITE VARCHAR(12)	PAYLOAD VARCHAR(61)	PAYLOAD_MASS__KG_ INTEGER	ORBIT VARCHAR(11)	CUSTOMER VARCHAR(57)
CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO
CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)
VAFB SLC-4E	CASSIOPE	500	Polar LEO	MDA
CCAFS LC-40	SES-8	3170	GTO	SES
CCAFS LC-40	Thaicom 6	3325	GTO	Thaicom
CCAFS LC-40	SpaceX CRS-3	2296	LEO (ISS)	NASA (CRS)
CCAFS LC-40	OG2 Mission 1 6 Orbcomm-OG2 satellites	1316	LEO	Orbcomm

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&utr 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
My Computer
Amazon S3
Cloud Object Storage

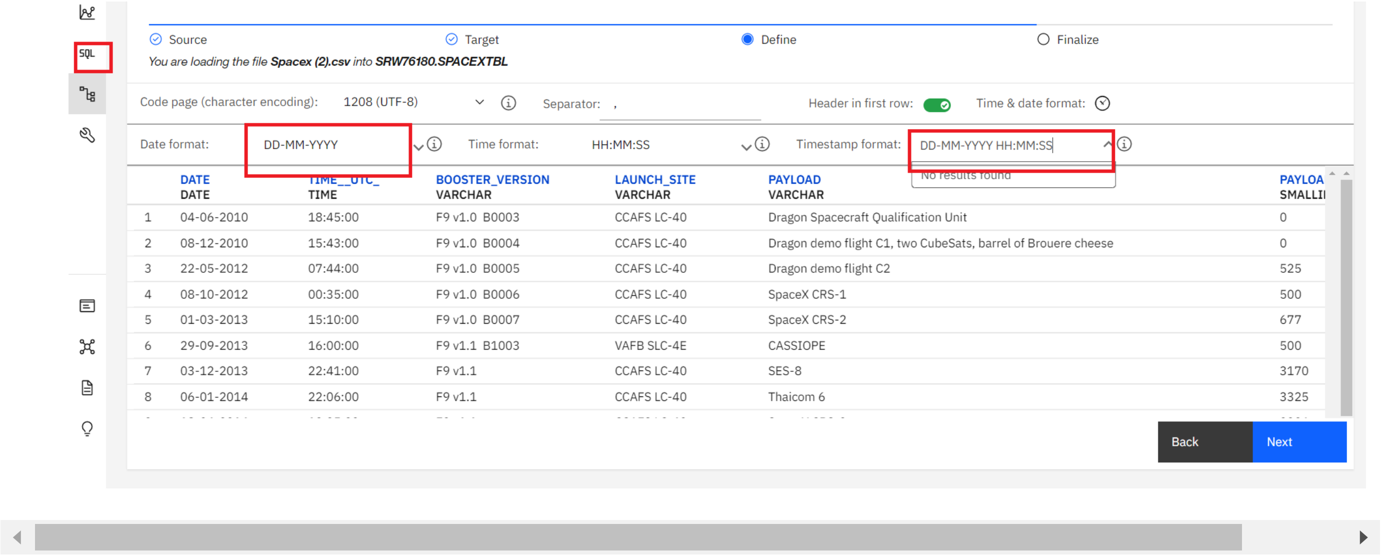
You are loading the file

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

File selection

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 [5]:

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

```
Requirement already satisfied: sqlalchemy==1.3.9 in c:\users\sunhi\anacond
a3\lib\site-packages (1.3.9)
Requirement already satisfied: ibm_db_sa in c:\users\sunhi\anaconda3\lib\s
ite-packages (0.3.7)
Requirement already satisfied: sqlalchemy>=0.7.3 in c:\users\sunhi\anacond
a3\lib\site-packages (from ibm_db_sa) (1.3.9)
Requirement already satisfied: ibm-db>=2.0.0 in c:\users\sunhi\anaconda3\l
ib\site-packages (from ibm_db_sa) (3.1.1)
Requirement already satisfied: ipython-sql in c:\users\sunhi\anaconda3\lib
\site-packages (0.4.0)
Requirement already satisfied: sqlalchemy>=0.6.7 in c:\users\sunhi\anacond
a3\lib\site-packages (from ipython-sql) (1.3.9)
Requirement already satisfied: ipython>=1.0 in c:\users\sunhi\anaconda3\li
b\site-packages (from ipython-sql) (7.12.0)
Requirement already satisfied: sqlparse in c:\users\sunhi\anaconda3\lib\si
te-packages (from ipython-sql) (0.4.2)
Requirement already satisfied: prettytable<1 in c:\users\sunhi\anaconda3\l
ib\site-packages (from ipython-sql) (0.7.2)
Requirement already satisfied: six in c:\users\sunhi\anaconda3\lib\site-pa
ckages (from ipython-sql) (1.14.0)
Requirement already satisfied: ipython-genutils>=0.1.0 in c:\users\sunhi\a
naconda3\lib\site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.
0 in c:\users\sunhi\anaconda3\lib\site-packages (from ipython>=1.0->ipytho
n-sql) (3.0.3)
Requirement already satisfied: traitlets>=4.2 in c:\users\sunhi\anaconda3
\lib\site-packages (from ipython>=1.0->ipython-sql) (4.3.3)
Requirement already satisfied: jedi>=0.10 in c:\users\sunhi\anaconda3\lib
\site-packages (from ipython>=1.0->ipython-sql) (0.14.1)
Requirement already satisfied: backcall in c:\users\sunhi\anaconda3\lib\si
te-packages (from ipython>=1.0->ipython-sql) (0.1.0)
Requirement already satisfied: pygments in c:\users\sunhi\anaconda3\lib\si
te-packages (from ipython>=1.0->ipython-sql) (2.5.2)
Requirement already satisfied: decorator in c:\users\sunhi\anaconda3\lib\s
ite-packages (from ipython>=1.0->ipython-sql) (4.4.1)
Requirement already satisfied: pickleshare in c:\users\sunhi\anaconda3\lib
\site-packages (from ipython>=1.0->ipython-sql) (0.7.5)
Requirement already satisfied: colorama; sys_platform == "win32" in c:\use
rs\sunhi\anaconda3\lib\site-packages (from ipython>=1.0->ipython-sql) (0.
4.3)
Requirement already satisfied: setuptools>=18.5 in c:\users\sunhi\anaconda
3\lib\site-packages (from ipython>=1.0->ipython-sql) (45.2.0.post20200210)
Requirement already satisfied: wcwidth in c:\users\sunhi\anaconda3\lib\sit
e-packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=1.
0->ipython-sql) (0.1.8)
Requirement already satisfied: parso>=0.5.0 in c:\users\sunhi\anaconda3\li
b\site-packages (from jedi>=0.10->ipython>=1.0->ipython-sql) (0.5.2)
```

Connect to the database

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

In [6]:

```
import ibm_db
import ibm_db_sa
import sqlalchemy
```

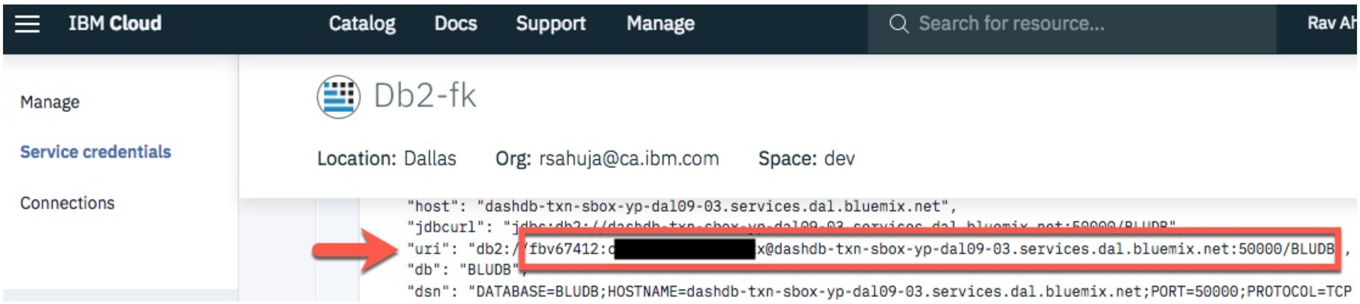
In [7]:

```
%load_ext sql
```

```
The sql extension is already loaded. To reload it, use:
  %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://



Db2-fk

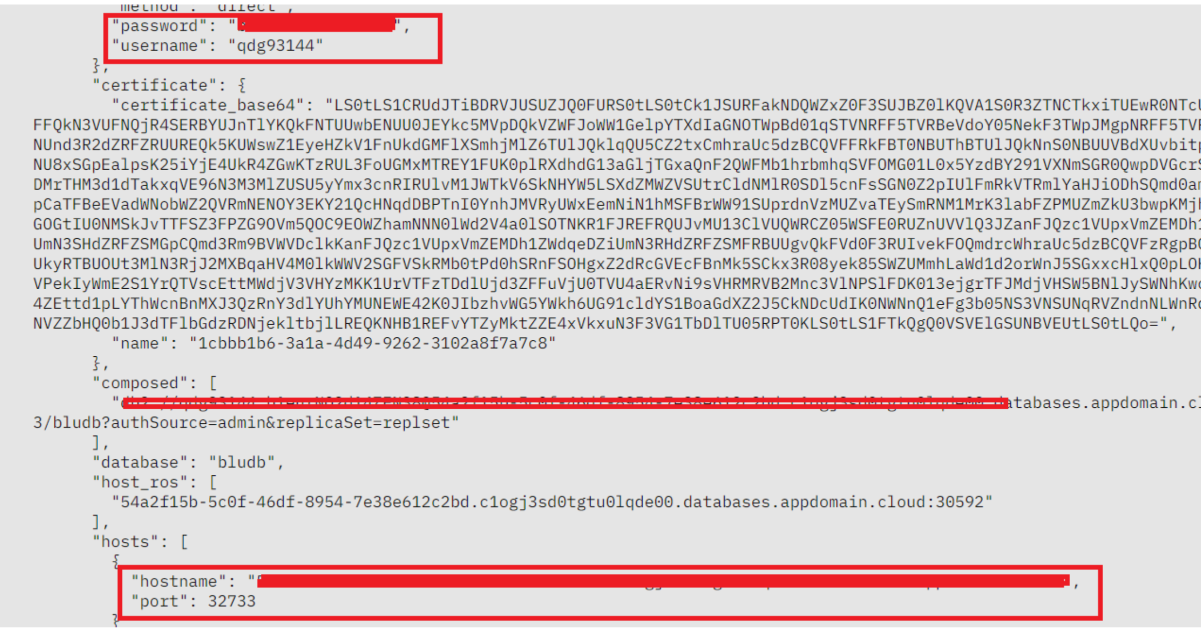
Location: Dallas Org: rsahuja@ca.ibm.com Space: dev

```
"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.c
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 [11]:

```
#%sql ibm_db_sa://
%sql ibm_db_sa://zrq46142:AyZwKcF1lNYdkuBH@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90
l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb?security=SSL
```

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 [13]:

```
%sql SELECT Distinct LAUNCH_SITE FROM SPACEX
```

```
 * ibm_db_sa://zrq46142:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l0
8kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

Out[13]:

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 'KSC'

In [14]:

```
%sql SELECT * FROM SPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://zrq46142:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l0
8kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

Out[14]:

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	Sp
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)	| (C
2012-08-10	00:35:00	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)	| (
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	

Task 3

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

In [15]:

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEX WHERE CUSTOMER='NASA (CRS)'
```

```
 * ibm_db_sa://zrq46142:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l0
8kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

Out[15]:

1
22007

Task 4

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

In [16]:

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEX WHERE BOOSTER_VERSION='F9 v1.1'
```

```
 * ibm_db_sa://zrq46142:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l0
8kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

Out[16]:

1
3676

Task 5

List the date where the first succesful landing outcome in drone ship was acheived.

Hint:Use min function

In [17]:

```
%sql SELECT min(DATE) FROM SPACEX WHERE LANDING__OUTCOME='Success (ground pad)'
```

```
 * ibm_db_sa://zrq46142:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l0
8kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

Out[17]:

1
2017-01-05

Task 6

List the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000

In [18]:

```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE PAYLOAD_MASS__KG_ between 4000 and 6000 A
ND LANDING__OUTCOME='Success (drone ship)'
```

```
 * ibm_db_sa://zrq46142:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l0
8kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

Out[18]:

booster_version
F9 FT B1022
F9 FT B1031.2

Task 7

List the total number of successful and failure mission outcomes

In [19]:

```
%sql SELECT COUNT(*) FROM SPACEX WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTC
OME LIKE '%Failure%'
```

```
 * ibm_db_sa://zrq46142:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l0
8kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

Out[19]:

1
45

Task 8

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

In [20]:

```
%sql SELECT BOOSTER_VERSION FROM SPACEX WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_M
ASS__KG_) FROM SPACEXTBL)
```

```
 * ibm_db_sa://zrq46142:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l0
8kqb1od8lcg.databases.appdomain.cloud:31505/bludb
(ibm_db_dbi.ProgrammingError) ibm_db_dbi::ProgrammingError: SQLNumResultCo
ls failed: [IBM][CLI Driver][DB2/LINUXX8664] SQL0204N  "ZRQ46142.SPACEXTB
L" is an undefined name.  SQLSTATE=42704\r SQLCODE=-204
[SQL: SELECT BOOSTER_VERSION FROM SPACEX WHERE PAYLOAD_MASS__KG_ = (SELECT
MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)]
(Background on this error at: http://sqlalche.me/e/f405)
```

Task 9

List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

In [21]:

```
%sql SELECT TO_CHAR(TO_DATE(MONTH("DATE"), 'MM'), 'MONTH') AS MONTH_NAME, \
    LANDING__OUTCOME AS LANDING__OUTCOME, \
    BOOSTER_VERSION AS BOOSTER_VERSION, \
    LAUNCH_SITE AS LAUNCH_SITE \
    FROM SPACEX WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015%
'
```

```
 * ibm_db_sa://zrq46142:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l0
8kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

Out[21]:

month_name	landing__outcome	booster_version	launch_site
OCTOBER	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40

Task 10

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

In [22]:

```
%sql SELECT "DATE", COUNT(LANDING__OUTCOME) as COUNT FROM SPACEX \
    WHERE "DATE" BETWEEN '2010-06-04' and '2017-03-20' AND LANDING__OUTCOME LIKE '%Succ
ess%' \
    GROUP BY "DATE" \
    ORDER BY COUNT(LANDING__OUTCOME) DESC
```

```
 * ibm_db_sa://zrq46142:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l0
8kqb1od8lcg.databases.appdomain.cloud:31505/bludb
Done.
```

Out[22]:

DATE	COUNT
2016-06-05	1
2016-08-04	1
2017-01-05	1
2017-03-06	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/Module%203/LAB-String_Patterns_Sorting_Grouping.md.html?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utr SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01)
- 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&utr SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01)
- 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&utr SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01)
- 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&utr 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&utr SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01)

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