# SpaceX EDX SQL Server

**The dataset used to be queried can be found below.**

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)

## Connect to the database

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

In [1]:
```
!pip install sqlserver
```

```
/opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages/secretstorage/dhcrypto.py:16: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
/opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages/secretstorage/util.py:25: CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes instead
  from cryptography.utils import int_from_bytes
Collecting sqlserver
  Downloading sqlserver-0.0.4.tar.gz (2.9 kB)
Requirement already satisfied: pyodbc in /opt/conda/envs/Python-3.7-main/lib/python3.7/site-packages (from sqlserver) (4.0.0-unsupported)
Building wheels for collected packages: sqlserver
  Building wheel for sqlserver (setup.py) ... done
  Created wheel for sqlserver: filename=sqlserver-0.0.4-py3-none-any.whl size=2983 sha256=d7a6445a8f8724565aca45a1aac76d097e1ac4238c4292e5a5c8c021d9a4861f
  Stored in directory: /tmp/wsuser/.cache/pip/wheels/81/17/65/2e9e92ba3f82544a877f975582a7a2778a9d53b43b7478ba49
Successfully built sqlserver
Installing collected packages: sqlserver
Successfully installed sqlserver-0.0.4
```

## Initialization

the .sqlserver() object parameters stands for (ip,portnumber,databasename,username,password)

In [7]:
```
import sqlserver as ss
db = ss.sqlserver('localhost','1433','SpaceX','admin','admin')
```

### Display the names of the unique launch sites in the space mission

In [8]:
```
db.GetRecordsOfColumn('select DISTINCT Launch_Site from tblSpaceX','Launch_Site')
```

Out[8]:
```
['CCAFS LC-40', 'CCAFS SLC-40', 'CCAFSSLC-40', 'KSC LC-39A', 'VAFB SLC-4E']
```

### Display 5 records where launch sites begin with the string 'KSC'

In [41]:
```
import pyodbc
import pandas as pd
import numpy as np
conn = pyodbc.connect('Driver={SQL Server};'
                                    'Server=localhost;'
                                    'Database=SpaceX;'
                                    'User ID=admin;Password=admin;')
cursor = conn.cursor()

cursor.execute("select TOP 5 * from tblSpaceX WHERE Launch_Site LIKE 'KSC%'")
columns = [column[0] for column in cursor.description]
results = []
for row in cursor.fetchall():
     results.append(dict(zip(columns, row)))

df = pd.DataFrame.from_dict(results)
df
```

Out[41]:

| | Date | Time_UTC | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS_KG_ | Orbit |
|---|---|---|---|---|---|---|---|
| **0** | 19-02-2017 | 2021-07-02 14:39:00.0000000 | F9 FT B1031.1 | KSC LC-39A | SpaceX CRS-10 | 2490 | LEO (ISS) |
| **1** | 16-03-2017 | 2021-07-02 06:00:00.0000000 | F9 FT B1030 | KSC LC-39A | EchoStar 23 | 5600 | GTO |
| **2** | 30-03-2017 | 2021-07-02 22:27:00.0000000 | F9 FT B1021.2 | KSC LC-39A | SES-10 | 5300 | GTO |
| **3** | 01-05-2017 | 2021-07-02 11:15:00.0000000 | F9 FT B1032.1 | KSC LC-39A | NROL-76 | 5300 | LEO |
| **4** | 15-05-2017 | 2021-07-02 23:21:00.0000000 | F9 FT B1034 | KSC LC-39A | Inmarsat-5 F4 | 6070 | GTO |

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

In [57]:
```
TPM = db.GetRecordsOfColumn("select SUM(PAYLOAD_MASS_KG_) TotalPayloadMass fro
m tblSpaceX where Customer = 'NASA (CRS)'",'TotalPayloadMass')
ndf= pd.DataFrame(TPM)
ndf.columns = ['Total Payload Mass']
ndf
```

Out[57]:

| | Total Payload Mass |
|---|---|
| **0** | 45596 |

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

In [62]:
```
APM = db.GetRecordsOfColumn("select AVG(PAYLOAD_MASS_KG_) AveragePayloadMass f
rom tblSpaceX where Booster_Version = 'F9 v1.1'",'AveragePayloadMass')
ndf= pd.DataFrame(APM)
ndf.columns = ['Average Payload Mass']
ndf
```

Out[62]:

| | Average Payload Mass |
|---|---|
| **0** | 2928 |

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

In [64]:
```
SLO = db.GetRecordsOfColumn("select MIN(Date) SLO from tblSpaceX where Landing
_Outcome = 'Success (drone ship)'",'SLO')
ndf= pd.DataFrame(SLO)
ndf.columns = ['Date which first Successful landing outcome in drone ship was
 acheived.']
ndf
```

Out[64]:

| | Date which first Successful landing outcome in drone ship was acheived. |
|---|---|
| **0** | 06-05-2016 |

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

In [69]:
```
SLO = db.GetRecordsOfColumn("select Booster_Version from tblSpaceX where Landi
ng_Outcome = 'Success (ground pad)' AND Payload_MASS_KG_ > 4000 AND Payload_MA
SS_KG_ < 6000",'Booster_Version')
ndf= pd.DataFrame(SLO)
ndf.columns = ['Date which first Successful landing outcome in drone ship was
 acheived.']
ndf
```

Out[69]:

| | Date which first Successful landing outcome in drone ship was acheived. |
|---|---|
| **0** | F9 FT B1032.1 |
| **1** | F9 B4 B1040.1 |
| **2** | F9 B4 B1043.1 |

**List the total number of successful and failure mission outcomes**

In [84]:
```
conn = pyodbc.connect('Driver={SQL Server};'
                                  'Server=localhost;'
                                  'Database=SpaceX;'
                                  'User ID=admin;Password=admin;')
cursor = conn.cursor()

cursor.execute("SELECT(SELECT Count(Mission_Outcome) from tblSpaceX where Miss
ion_Outcome LIKE '%Success%') as Successful_Mission_Outcomes,(SELECT Count(Mis
sion_Outcome) from tblSpaceX where Mission_Outcome LIKE '%Failure%') as Failur
e_Mission_Outcomes")
columns = [column[0] for column in cursor.description]
results = []
for row in cursor.fetchall():
     results.append(dict(zip(columns, row)))

df = pd.DataFrame.from_dict(results)
df
```

Out[84]:

| | Successful_Mission_Outcomes | Failure_Mission_Outcomes |
|---|---|---|
| **0** | 100 | 1 |

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

In [94]:
```
conn = pyodbc.connect('Driver={SQL Server};'
                      'Server=localhost;'
                      'Database=SpaceX;'
                      'User ID=admin;Password=admin;')
cursor = conn.cursor()

cursor.execute("SELECT DISTINCT Booster_Version, MAX(PAYLOAD_MASS_KG_) AS [Max
imum Payload Mass] FROM tblSpaceX GROUP BY Booster_Version ORDER BY [Maximum P
ayload Mass] DESC")
columns = [column[0] for column in cursor.description]
results = []
for row in cursor.fetchall():
    results.append(dict(zip(columns, row)))

df = pd.DataFrame.from_dict(results)
df
```

Out[94]:

| | Booster_Version | Maximum Payload Mass |
|---|---|---|
| **0** | F9 B5 B1048.4 | 15600 |
| **1** | F9 B5 B1048.5 | 15600 |
| **2** | F9 B5 B1049.4 | 15600 |
| **3** | F9 B5 B1049.5 | 15600 |
| **4** | F9 B5 B1049.7 | 15600 |
| **...** | ... | ... |
| **92** | F9 v1.1 B1003 | 500 |
| **93** | F9 FT B1038.1 | 475 |
| **94** | F9 B4 B1045.1 | 362 |
| **95** | F9 v1.0 B0003 | 0 |
| **96** | F9 v1.0 B0004 | 0 |

97 rows × 2 columns

**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 [96]:
```
conn = pyodbc.connect('Driver={SQL Server};'
                                    'Server=localhost;'
                                    'Database=SpaceX;'
                                    'User ID=admin;Password=admin;')
cursor = conn.cursor()

cursor.execute("SELECT   DateName( month , DateAdd( month , MONTH(CONVERT(dat
e,Date, 105)) , 0 ) - 1 )  as Month, Booster_Version, Launch_Site, Landing_Out
come FROM tblSpaceX WHERE  (Landing_Outcome LIKE N'%Success%') AND YEAR(CONVER
T(date,Date, 105)) = '2017'")
columns = [column[0] for column in cursor.description]
results = []
for row in cursor.fetchall():
     results.append(dict(zip(columns, row)))

df = pd.DataFrame.from_dict(results)
df
```

Out[96]:

| | Month | Booster_Version | Launch_Site | Landing_Outcome |
|---|---|---|---|---|
| **0** | January | F9 FT B1029.1 | VAFB SLC-4E | Success (drone ship) |
| **1** | February | F9 FT B1031.1 | KSC LC-39A | Success (ground pad) |
| **2** | March | F9 FT B1021.2 | KSC LC-39A | Success (drone ship) |
| **3** | May | F9 FT B1032.1 | KSC LC-39A | Success (ground pad) |
| **4** | June | F9 FT B1035.1 | KSC LC-39A | Success (ground pad) |
| **5** | June | F9 FT B1029.2 | KSC LC-39A | Success (drone ship) |
| **6** | June | F9 FT B1036.1 | VAFB SLC-4E | Success (drone ship) |
| **7** | August | F9 B4 B1039.1 | KSC LC-39A | Success (ground pad) |
| **8** | August | F9 FT B1038.1 | VAFB SLC-4E | Success (drone ship) |
| **9** | September | F9 B4 B1040.1 | KSC LC-39A | Success (ground pad) |
| **10** | October | F9 B4 B1041.1 | VAFB SLC-4E | Success (drone ship) |
| **11** | October | F9 FT B1031.2 | KSC LC-39A | Success (drone ship) |
| **12** | October | F9 B4 B1042.1 | KSC LC-39A | Success (drone ship) |
| **13** | December | F9 FT B1035.2 | CCAFS SLC-40 | Success (ground pad) |

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

In [90]:
```
sl = db.GetRecordsOfColumn("SELECT COUNT(Landing_Outcome) AS sl FROM dbo.tblSpaceX WHERE (Landing_Outcome LIKE '%Success%') AND (Date >'04-06-2010') AND (Date < '20-03-2017')",'sl')

ndf= pd.DataFrame(sl)
ndf.columns = ['Successful Landing Outcomes Between 2010-06-04 and 2017-03-20']
ndf
```

Out[90]:

| | Successful Landing Outcomes Between 2010-06-04 and 2017-03-20 |
|---|---|
| **0** | 34 |