



## IBM Developer SKILLS NETWORK

### 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 whereas 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 \(https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module\\_2/data/Spacex.csv?utm\\_medium=Exinfluencer&utm\\_source=Exinfluencer&utm\\_content=000026UJ&utm\\_term=10006555&utm\\_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMD�0321ENSkillsNetwork26802033-2021-01-01\)](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-SkillsNetworkCoursesIBMD�0321ENSkillsNetwork26802033-2021-01-01)

## 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

Source Target Define Finalize

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

Select a load target

Refresh

| Schema             | Table                    | Create a new Table |
|--------------------|--------------------------|--------------------|
| Find a schema      | Find a table in QWP24135 | SPACEXTBL          |
| AUDIT              | ANNUAL_CROP_DATA         | Create             |
| DB2INST1           | BOARD                    |                    |
| ERRORSCHEMA Sample | BOOKSHOP                 |                    |
| IDAX               | BOOKSHOP_AUTHORDetails   |                    |
| QWP24135           | CAR_SALES                |                    |
| SQL15777           | CAR_SALES_DATA           |                    |

Back Next

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 PAYLOADMASS\\_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 instruction in this [link \(https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Labs\\_Coursera\\_V5/labs/Lab%20-%20Sign%20up%20for%20IBM%20Cloud%20-%20Create%20Db2%20service%20instance%20-%20Get%20started%20with%20the%20Db2%20console/instructional-labs.md.html?utm\\_medium=Exinfluencer&utm\\_source=Exinfluencer&utm\\_content=000026UJ&utm\\_term=10006555&utm\\_ic SkillsNetwork-Channel-SkillsNetworkCoursesIBMD50321ENSkillsNetwork26802033-2021-01-01\)](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_ic SkillsNetwork-Channel-SkillsNetworkCoursesIBMD50321ENSkillsNetwork26802033-2021-01-01) 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.

- Once done select the schema and load the file.

| DATE DATE | TIME 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/envs/python/lib/python3.6/site-packages (1.3.9)

Requirement already satisfied: ibm\_db\_sa in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (0.3.3)

Requirement already satisfied: sqlalchemy>=0.7.3 in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ibm\_db\_sa) (1.3.9)

Requirement already satisfied: ipython-sql in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (0.3.9)

Requirement already satisfied: ipython>=1.0 in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython-sql) (7.16.1)

Requirement already satisfied: sqlparse in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython-sql) (0.4.1)

Requirement already satisfied: prettytable in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython-sql) (2.1.0)

Requirement already satisfied: sqlalchemy>=0.6.7 in /home/jupyterlab/conda/envs/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: ipython-genutils>=0.1.0 in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython-sql) (0.2.0)

Requirement already satisfied: decorator in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython>=1.0->ipython-sql) (4.4.2)

Requirement already satisfied: backcall in /home/jupyterlab/conda/envs/python/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 in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython>=1.0->ipython-sql) (3.0.19)

Requirement already satisfied: pexpect; sys\_platform != "win32" in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython>=1.0->ipython-sql) (4.8.0)

Requirement already satisfied: pygments in /home/jupyterlab/conda/envs/python/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/python/lib/python3.6/site-packages (from ipython>=1.0->ipython-sql) (0.17.2)

Requirement already satisfied: pickleshare in /home/jupyterlab/conda/envs/python/lib/python3.6/site-packages (from ipython>=1.0->ipython-sql) (0.7.5)

Requirement already satisfied: setuptools>=18.5 in /home/jupyterlab/conda/envs/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 prettytable->ipython-sql) (4.6.1)

Requirement already satisfied: ptyprocess>=0.5 in /home/jupyterlab/conda/envs/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->ipython-sql) (0.7.1)

Requirement already satisfied: zipp>=0.5 in /home/jupyterlab/conda/envs/python/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 [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://`

IBM Cloud Catalog Docs Support Manage Search for resource... Rav Ah

Manage

Service credentials

Connections

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: [redacted]@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": "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURFakNDQWZxZ0F3SUJBZ0lKQVA1S0R3ZTNCTkxiTUEwR0NTc1FFQkN3VUfNqjR4SERBYUJnT1YkQkFNTUwEUNU0JEYkc5MVpDQkVZWJfJ0wW1GelpYTXdIaGNOTWpBd01qSTVNRF5TVRBeVdoY05NekF3TWpJMGpNRFF5TVI
NUnd3R2dZRFZRUUREQk5KUwswZ1EYehZkV1FnUkdGMF1XSmhjM1Z6TU1JQk1lQU5CZ2txCmhraUc5dzBCQVFFRkFT0NBUTB0TjU1JQkNnS0NB0UUVBdXUvbitj
NU8xSGpEalpsK251YjE4UkR4ZGwKTzRUL3FoUGMxMTREY1FUK0p1RXdhG13aG1jTGxaQnF2QWFMb1hzbmhqSVFOMG01L0x5YzdBY291VXNmSGR0QwpDVGcr!
DMrTHM3d1dTakxqVE96N3M1ZUSU5yYmx3cnRIRUlVMTJWTKV6SkNHYW5LSXZdZWZVSUtrCldNM1R0SD15cnFsSGN0Z2pIU1FmRkVTRm1YaHJiODhSQmd0ar
pCaTFBeEadVwNobWZ2QVRmNEN0Y3EKY21QcHNqdDBPTnI0YnhJMVRYUWxEemNiN1hMSFBrWw91SUpidnVzMUZvaTEySmNM1MxK3labFZPMUZmZkU3bwpKMjI
G0GtIU0NMMSKJvTTF5Z3FPZG90Vm5QOC9EOWZhamNN01Wd2V4a01S0TNKR1FJREFRQUJvMU13ClVUQWRRC05WSE0RUZnUUVV1Q3JZanFJQzc1VUpXVmZEMDh:
UmN3SHdZRFZSMGpCQmd3Rm9BVWVDclKkanFJQzc1VUpXVmZEMDh1ZwdqeDZlUmN3RhdZRFZSMFRBUUgVqkFVd0F3RU1IvekF0QmdrcWhraUc5dzBCQVZFRgpB
UkyRTBU0U0t3M1N3RjJ2MkRqAHV4M01kWWV2SGFVSkrMb0tPd0hSRnFSOHgxZ2dRcGVEcFBnMk5SCkx3R08yek85SWZUMmhLaWd1d2orWnJ5SGxcH1xQ0pLOI
VPEkIyWmE2S1YrQTVScEttMwdjV3VHYzMKK1U1rVTFzTdd1Ujd3ZFfVjU0TVU4aERvNi9sVHRMRVB2Mnc3VlNPS1FDK013ejgrTFJmJdVHWS5BN1JySWNhKw
4ZEttD1pLYThWcnBnMXJ3QzRnY3d1YUhYMUENEWE42K0JibzhvWG5YWhk6UG91clYs1BoaGdXZ2J3cKNDcUdIK0NWNnQ1eFg3b05NS3VNSUNqRVZndnNLWnR
NVZZbH00b1J3dTF1bGdzRDNjek1tbj1LREQKNHB1REFvYTZyMktZE4xVkkxN3F3VG1TbD1TU05RPT0KLS0tLS1FTkQgQ0VSVE1GSUNBEU0tLS0tLQo=",
  "name": "1cbbb1b6-3a1a-4d49-9262-3102a8f7a7c8"
},
"composed": [
  "3/bludb?authSource=admin&replicaSet=rep1set"
],
"database": "bludb",
"host_ros": [
  "54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.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 [6]:

```
%sql ibm_db_sa://jgl54010:t5rx38n6nr%2Btnv8w@dashdb-txn-sbox-yp-dal09-08.services.dal.bluemix.net:50000/BLUDB?security=SSL
```

```
(ibm_db_dbi.OperationalError) ibm_db_dbi::OperationalError: [IBM][CLI Driver] SQL30081N A communication error has been detected. Communication protocol being used: "SSL". Communication API being used: "SOCKETS". Location where the error was detected: "". Communication function detecting the error: "sqlcc SSLSocketSetup". Protocol specific error code(s): "420", "**", "**". SQLSTATE =08001 SQLCODE=-30081
```

(Background on this error at: <http://sqlalche.me/e/e3q8>)

Connection info needed in SQLAlchemy format, example:

```
postgresql://username:password@hostname/dbname
```

```
or an existing connection: dict_keys(['ibm_db_sa://jgl54010:**@dashdb-txn-sbox-yp-dal09-08.services.dal.bluemix.net:50000/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 [7]:

```
%sql select DISTINCT(UCASE(LAUNCH_SITE)) from SPACEXTBL
```

```
* ibm_db_sa://jgl54010:**@dashdb-txn-sbox-yp-dal09-08.services.dal.bluemix.net:50000/BLUDB
Done.
```

Out[7]:

| 1            |
|--------------|
| 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 [18]:

```
%sql select LAUNCH_SITE from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5

* ibm_db_sa://jgl54010:***@dashdb-txn-sbox-yp-dal09-08.services.dal.ibm.com
net:50000/BLUDB
Done.
```

Out[18]:

| launch_site |
|-------------|
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |
| CCAFS LC-40 |

### Task 3

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

In [24]:

```
%sql select SUM(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER='NASA (CRS) '

* ibm_db_sa://jgl54010:***@dashdb-txn-sbox-yp-dal09-08.services.dal.ibm.com
net:50000/BLUDB
Done.
```

Out[24]:

| 1     |
|-------|
| 45596 |

### Task 4

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

In [25]:

```
%sql select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION='F9 v1.1'

* ibm_db_sa://jgl54010:***@dashdb-txn-sbox-yp-dal09-08.services.dal.ibm.com
net:50000/BLUDB
Done.
```

Out[25]:

| 1           |
|-------------|
| 2928.400000 |

## Task 5

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

*Hint: Use min function*

In [46]:

```
%sql select MIN (DATE) from SPACEXTBL where LANDING__OUTCOME = 'Success (ground pad)'

* ibm_db_sa://jgl54010:***@dashdb-txn-sbox-yp-dal09-08.services.dal.ibm.com
net:50000/BLUDB
Done.
```

Out[46]:

| 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 [103]:

```
%sql select BOOSTER_VERSION, PAYLOAD_MASS__KG_ from SPACEXTBL where LANDING__OUTCOME = 'Success (drone ship)' and PAYLOAD_MASS__KG_ \
BETWEEN 4000 and 6000
```

```
* ibm_db_sa://jgl54010:***@dashdb-txn-sbox-yp-dal09-08.services.dal.ibm.com:50000/BLUDB
Done.
```

Out[103]:

| booster_version | payload_mass__kg_ |
|-----------------|-------------------|
| F9 FT B1022     | 4696              |
| F9 FT B1026     | 4600              |
| F9 FT B1021.2   | 5300              |
| F9 FT B1031.2   | 5200              |

## Task 7

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

In [58]:

```
%sql select COUNT(*) from SPACEXTBL where (MISSION_OUTCOME) LIKE 'Success%'
# %sql select COUNT(*) from SPACEXTBL where (MISSION_OUTCOME) LIKE 'Failure%'
```

```
* ibm_db_sa://jgl54010:***@dashdb-txn-sbox-yp-dal09-08.services.dal.ibm.com:50000/BLUDB
Done.
```

Out[58]:

|     |
|-----|
| 1   |
| 100 |

## Task 8

**List the names of the booster\_versions which have carried the maximum payload mass. Use a subquery**

In [61]:

```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select Max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

```
* ibm_db_sa://jgl54010:***@dashdb-txn-sbox-yp-dal09-08.services.dal.ibm.com
net:50000/BLUDB
Done.
```

Out[61]:

| 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 failed landing\_outcomes in drone ship, their booster versions, and launch site names for in year 2015*

In [70]:

```
%sql select BOOSTER_VERSION, LAUNCH_SITE, LANDING__OUTCOME, DATE from SPACEXTBL where LANDING__OUTCOME = 'Failure (drone ship)' and YEAR(DATE) < 2016
```

```
* ibm_db_sa://jgl54010:***@dashdb-txn-sbox-yp-dal09-08.services.dal.ibm.com
net:50000/BLUDB
Done.
```

Out[70]:

| booster_version | launch_site | landing__outcome | DATE |
|-----------------|-------------|------------------|------|
|-----------------|-------------|------------------|------|

|               |             |                      |            |
|---------------|-------------|----------------------|------------|
| F9 v1.1 B1012 | CCAFS LC-40 | Failure (drone ship) | 2015-01-10 |
|---------------|-------------|----------------------|------------|

|               |             |                      |            |
|---------------|-------------|----------------------|------------|
| F9 v1.1 B1015 | CCAFS LC-40 | Failure (drone ship) | 2015-04-14 |
|---------------|-------------|----------------------|------------|

## 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 [102]:

```
%sql select count(LANDING__OUTCOME), LANDING__OUTCOME from SPACEXTBL where DATE BETWEEN '2010-06-04' and '2017-03-20' GROUP BY LANDING__OUTCOME \
ORDER BY Count DESC
```

```
* ibm_db_sa://jgl54010:***@dashdb-txn-sbox-yp-dal09-08.services.dal.ibm.com:50000/BLUDB
Done.
```

Out[102]:

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

## Reference Links

- [Hands-on Lab : String Patterns, Sorting and Grouping \(https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Labs\\_Coursera\\_V5/labs/Lab%20-%20String%20Patterns%20-%20Sorting%20-%20Grouping/instructional-labs.md.html?utm\\_medium=Exinfluencer&utm\\_source=Exinfluencer&utm\\_content=000026UJ&utm\\_term=10006555&utm\\_ic SkillsNetwork-Channel-SkillsNetworkCoursesIBMD50321ENSkillsNetwork26802033-2021-01-01&origin=www.coursera.org\)](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_ic SkillsNetwork-Channel-SkillsNetworkCoursesIBMD50321ENSkillsNetwork26802033-2021-01-01&origin=www.coursera.org)
- [Hands-on Lab: Built-in functions \(https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Labs\\_Coursera\\_V5/labs/Lab%20-%20Built-in%20functions%20/Hands-on\\_Lab\\_Built-in\\_Functions.md.html?utm\\_medium=Exinfluencer&utm\\_source=Exinfluencer&utm\\_content=000026UJ&utm\\_term=10006555&utm\\_ic SkillsNetwork-Channel-SkillsNetworkCoursesIBMD50321ENSkillsNetwork26802033-2021-01-01&origin=www.coursera.org\)](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_ic SkillsNetwork-Channel-SkillsNetworkCoursesIBMD50321ENSkillsNetwork26802033-2021-01-01&origin=www.coursera.org)
- [Hands-on Lab : Sub-queries and Nested SELECT Statements \(https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Labs\\_Coursera\\_V5/labs/Lab%20-%20Sub-queries%20and%20Nested%20SELECTs%20/instructional-labs.md.html?utm\\_medium=Exinfluencer&utm\\_source=Exinfluencer&utm\\_content=000026UJ&utm\\_term=10006555&utm\\_ic SkillsNetwork-Channel-SkillsNetworkCoursesIBMD50321ENSkillsNetwork26802033-2021-01-01&origin=www.coursera.org\)](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_ic SkillsNetwork-Channel-SkillsNetworkCoursesIBMD50321ENSkillsNetwork26802033-2021-01-01&origin=www.coursera.org)
- [Hands-on Tutorial: Accessing Databases with SQL magic \(https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Module%205/DB0201EN-Week3-1-3-SQLmagic.ipynb?utm\\_medium=Exinfluencer&utm\\_source=Exinfluencer&utm\\_content=000026UJ&utm\\_term=10006555&utm\\_ic SkillsNetwork-Channel-SkillsNetworkCoursesIBMD50321ENSkillsNetwork26802033-2021-01-01\)](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_ic SkillsNetwork-Channel-SkillsNetworkCoursesIBMD50321ENSkillsNetwork26802033-2021-01-01)
- [Hands-on Lab: Analyzing a real World Data Set \(https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/Module%205/DB0201EN-Week3-1-4-Analyzing.ipynb?utm\\_medium=Exinfluencer&utm\\_source=Exinfluencer&utm\\_content=000026UJ&utm\\_term=10006555&utm\\_ic SkillsNetwork-Channel-SkillsNetworkCoursesIBMD50321ENSkillsNetwork26802033-2021-01-01\)](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_ic SkillsNetwork-Channel-SkillsNetworkCoursesIBMD50321ENSkillsNetwork26802033-2021-01-01)

## 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   |

© IBM Corporation 2021. All rights reserved.