 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

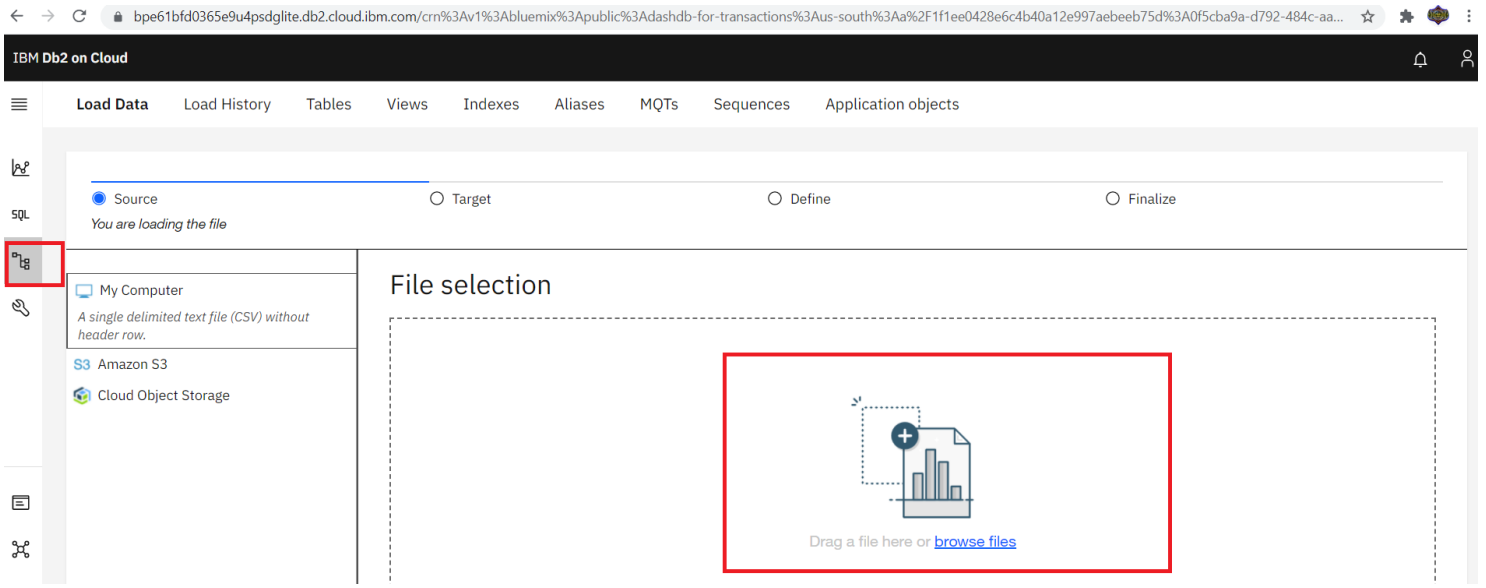
This assignment requires you to load the spacex dataset.

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):

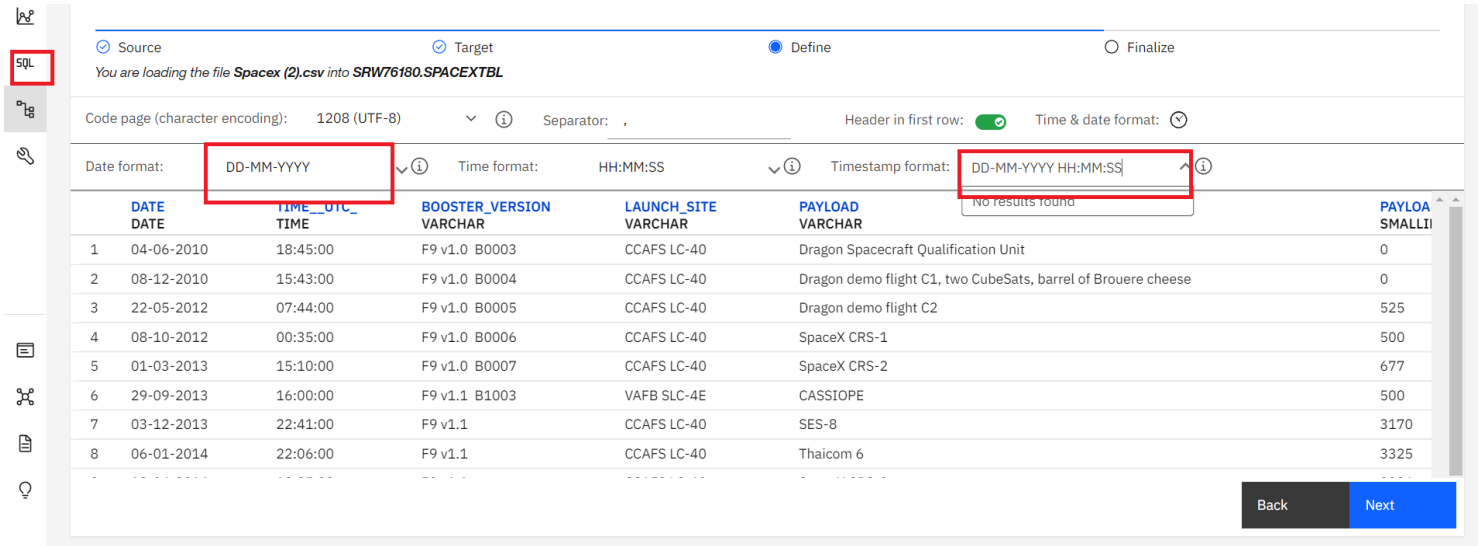
Spacex DataSet (https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2021-01-01)

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=NA-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.



- Once done select the schema andload the file.



In [1]:
```
!pip install sqlalchemy==1.3.9
!pip install ibm_db_sa
!pip install ipython-sql
```

```
Collecting sqlalchemy==1.3.9
  Downloading SQLAlchemy-1.3.9.tar.gz (6.0 MB)
     |████████████████████████████████| 6.0 MB 12.4 MB/s eta 0:00:01
Building wheels for collected packages: sqlalchemy
  Building wheel for sqlalchemy (setup.py) ... done
  Created wheel for sqlalchemy: filename=SQLAlchemy-1.3.9-cp39-cp39-linux_x86_64.whl size=1159949 sha256=9
4d3f811dbf04b3673d0bdb714ef7596f3d5516ab47635a30b071eb6f49b54c3
  Stored in directory: /tmp/wsuser/.cache/pip/wheels/5b/43/0d/de1699809f9e6aaa54a97275298fa07075cb19acc557
b18955
Successfully built sqlalchemy
Installing collected packages: sqlalchemy
  Attempting uninstall: sqlalchemy
    Found existing installation: SQLAlchemy 1.4.27
    Uninstalling SQLAlchemy-1.4.27:
      Successfully uninstalled SQLAlchemy-1.4.27
Successfully installed sqlalchemy-1.3.9
Requirement already satisfied: ibm_db_sa in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (0.3.7)
Requirement already satisfied: sqlalchemy>=0.7.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages
(from ibm_db_sa) (1.3.9)
Requirement already satisfied: ibm-db>=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (fr
om ibm_db_sa) (3.1.0)
Collecting ipython-sql
  Downloading ipython_sql-0.4.0-py3-none-any.whl (19 kB)
Requirement already satisfied: ipython-genutils>=0.1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-pa
ckages (from ipython-sql) (0.2.0)
Requirement already satisfied: sqlalchemy>=0.6.7 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages
(from ipython-sql) (1.3.9)
Collecting prettytable<1
  Downloading prettytable-0.7.2.zip (28 kB)
Requirement already satisfied: six in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ipython
-sql) (1.15.0)
Collecting sqlparse
  Downloading sqlparse-0.4.2-py3-none-any.whl (42 kB)
     |████████████████████████████████| 42 kB 3.5 MB/s  eta 0:00:01
Requirement already satisfied: ipython>=1.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (fro
m ipython-sql) (7.29.0)
Requirement already satisfied: decorator in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from i
python>=1.0->ipython-sql) (5.1.0)
Requirement already satisfied: setuptools>=18.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages
(from ipython>=1.0->ipython-sql) (58.0.4)
Requirement already satisfied: backcall in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ip
ython>=1.0->ipython-sql) (0.2.0)
Requirement already satisfied: pickleshare in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from
ipython>=1.0->ipython-sql) (0.7.5)
Requirement already satisfied: pygments in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ip
ython>=1.0->ipython-sql) (2.10.0)
Requirement already satisfied: pexpect>4.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from
ipython>=1.0->ipython-sql) (4.8.0)
Requirement already satisfied: traitlets>=4.2 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (f
rom ipython>=1.0->ipython-sql) (5.1.1)
Requirement already satisfied: matplotlib-inline in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages
(from ipython>=1.0->ipython-sql) (0.1.2)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0 in /opt/conda/envs/Python-3.9/
lib/python3.9/site-packages (from ipython>=1.0->ipython-sql) (3.0.20)
Requirement already satisfied: jedi>=0.16 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from
ipython>=1.0->ipython-sql) (0.18.0)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packag
es (from jedi>=0.16->ipython>=1.0->ipython-sql) (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages
(from pexpect>4.3->ipython>=1.0->ipython-sql) (0.7.0)
Requirement already satisfied: wcwidth in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pro
mpt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0->ipython>=1.0->ipython-sql) (0.2.5)
Building wheels for collected packages: prettytable
  Building wheel for prettytable (setup.py) ... done
  Created wheel for prettytable: filename=prettytable-0.7.2-py3-none-any.whl size=13714 sha256=3d2221c1e83
```

```
c173648fd58e1083572e9dd8d701f6f5dfa08c182ef3d59a52e2e
  Stored in directory: /tmp/wsuser/.cache/pip/wheels/75/f7/28/77a076f1fa8cbeda61aca712815d04d7a32435f04a26
a2dd7b
Successfully built prettytable
Installing collected packages: sqlparse, prettytable, ipython-sql
Successfully installed ipython-sql-0.4.0 prettytable-0.7.2 sqlparse-0.4.2
```

Connect to the database

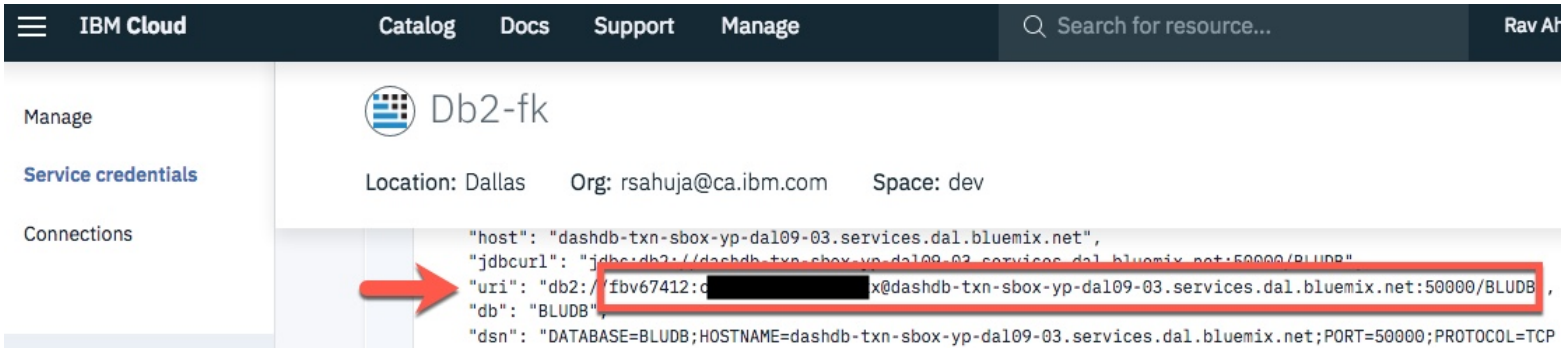
Let us first load the SQL extension and establish a connection with the database

In [5]:
```
%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://



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.

- 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 [6]:
```
%sql ibm_db_sa://jjt76647:[email  protected] (/cdn-cgi/l/email-protection)qnrk39u98g.databases.appdomain.cloud:30875/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 [9]:
```
%sql SELECT DISTINCT Launch_Site FROM JJT76647.SPACEXDATASET
```

```
 * ibm_db_sa://jjt76647:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[9]:

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 [6]:
```
%sql SELECT * FROM JJT76647.SPACEXDATASET WHERE launch_site LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://jjt76647:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[6]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landir
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	None	0	LEO	SpaceX	Success	Failu
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	None	0	LEO (ISS)	NASA (COTS) NRO	Success	Failu
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	None	525	LEO (ISS)	NASA (COTS)	Success	
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	None	500	LEO (ISS)	NASA (CRS)	Success	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	None	677	LEO (ISS)	NASA (CRS)	Success	

Task 3

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

In [8]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM JJT76647.SPACEXDATASET WHERE Customer='NASA (CRS)'
```

```
 * ibm_db_sa://jjt76647:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[8]:

1
45596

Task 4

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

In [10]:
```
%sql SELECT AVG(payload_mass__kg_) AS AVG_PAYLOAD_MASS FROM JJT76647.SPACEXDATASET WHERE booster_version='F9 v1.1';
```

```
 * ibm_db_sa://jjt76647:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[10]:

avg_payload_mass
2928

Task 5

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

Hint:Use min function

In [13]:
```
%sql SELECT MIN(DATE) AS first_successful_landing FROM JJT76647.SPACEXDATASET WHERE (LANDING__OUTCOME)='Success (ground pad)';
```

```
 * ibm_db_sa://jjt76647:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[13]:

first_successful_landing
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 [15]:
```
%sql SELECT DISTINCT(LANDING__OUTCOME) FROM JJT76647.SPACEXDATASET;
```

```
 * ibm_db_sa://jjt76647:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[15]:

landing__outcome
Controlled (ocean)
Failure
Failure (drone ship)
Failure (parachute)
No attempt
Precluded (drone ship)
Success
Success (drone ship)
Success (ground pad)
Uncontrolled (ocean)

Task 7

List the total number of successful and failure mission outcomes

In [16]:
```
%sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL FROM JJT76647.SPACEXDATASET GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://jjt76647:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[16]:

mission_outcome	total
Failure (in flight)	1
Success	99
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 [17]:
```
%sql SELECT MAX(PAYLOAD_MASS__KG_) FROM JJT76647.SPACEXDATASET;
```

```
 * ibm_db_sa://jjt76647:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[17]:

1
15600

In [21]:
```
%sql SELECT DISTINCT(BOOSTER_VERSION), (SELECT MAX(PAYLOAD_MASS__KG_) AS "maximum_payload_mass" FROM JJT76647.SPACEXDATASET) FROM JJT76647.SPACEXDATASET LIMIT 5;
```

```
 * ibm_db_sa://jjt76647:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[21]:

booster_version	maximum_payload_mass
F9 B4 B1039.2	15600
F9 B4 B1040.2	15600
F9 B4 B1041.2	15600
F9 B4 B1043.2	15600
F9 B4 B1039.1	15600

Task 9

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

In [22]:
```
%sql SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE, DATE FROM JJT76647.SPACEXDATASET WHERE LANDING__OUTCOME LIKE '%Failure (drone ship)%' AND (DATE LIKE '2015%') ;
```

```
 * ibm_db_sa://jjt76647:***@98538591-7217-4024-b027-8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:30875/bludb
Done.
```

Out[22]:

landing__outcome	booster_version	launch_site	DATE
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015-01-10
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	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

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=NA-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=NA-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=NA-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=NA-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=NA-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