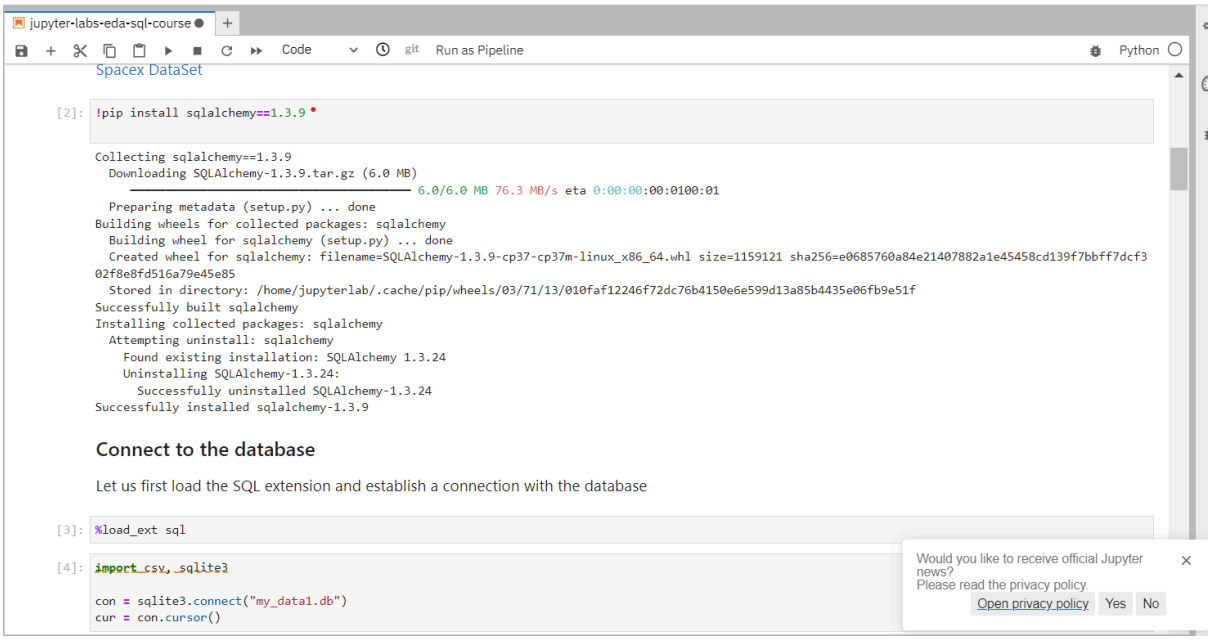
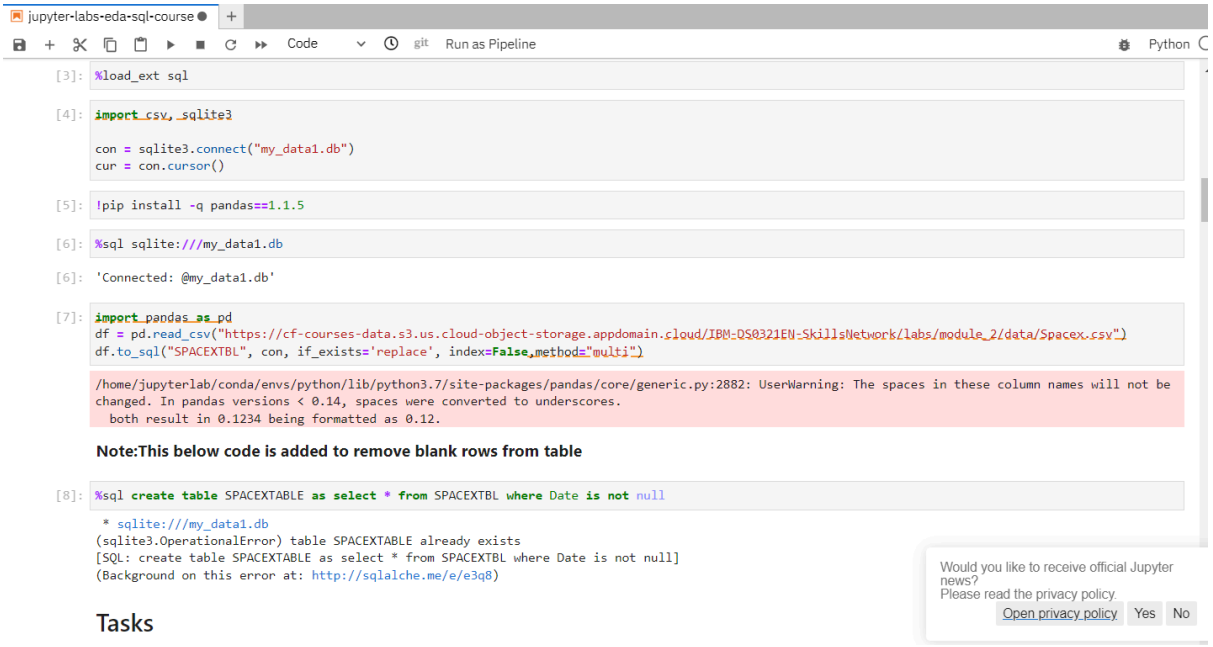
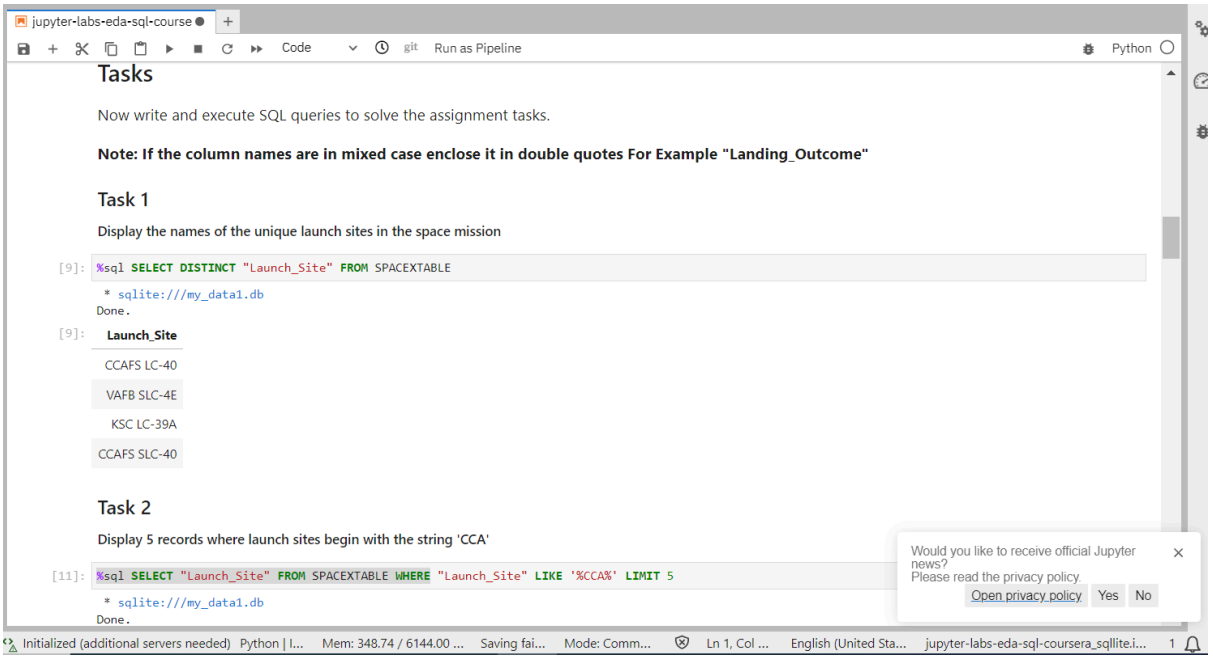
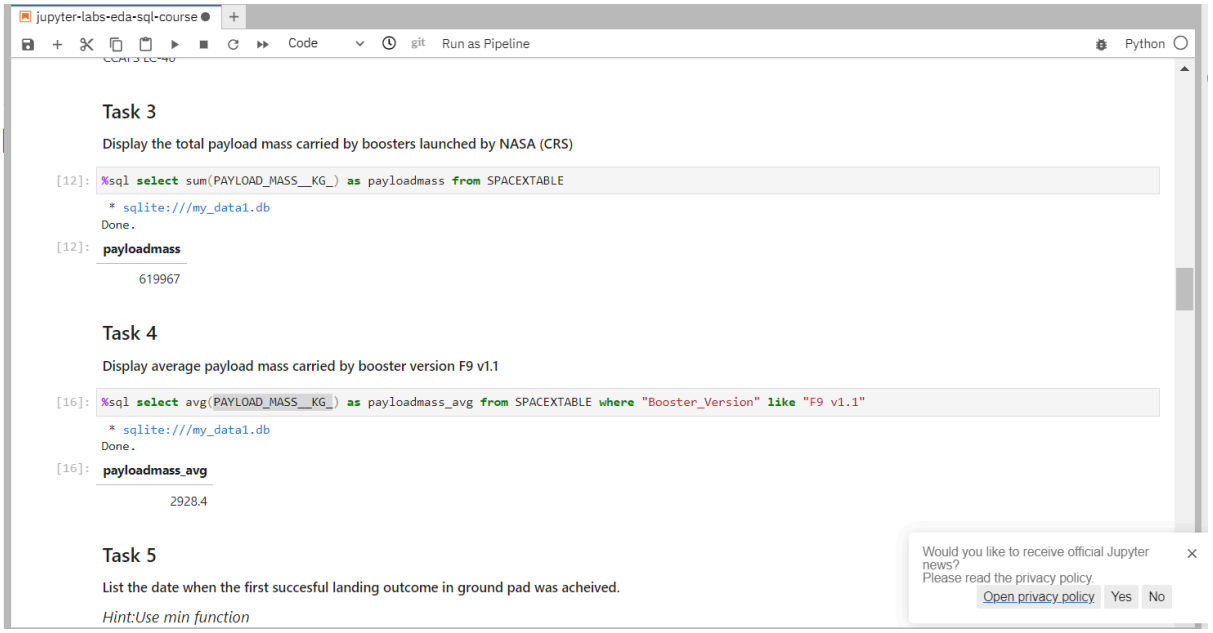
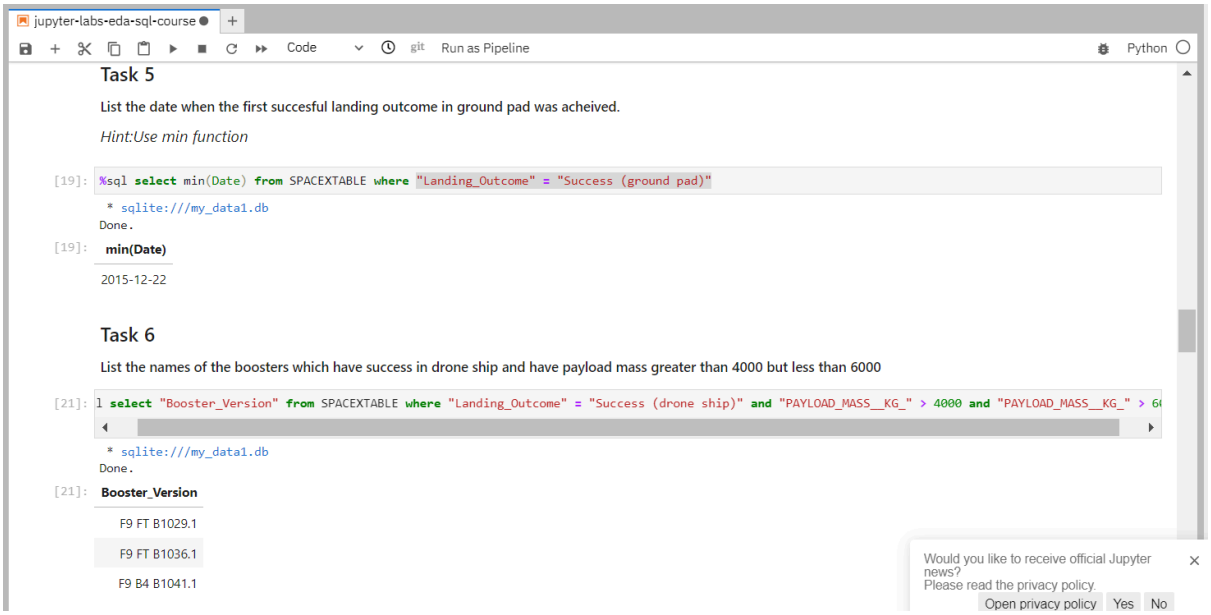
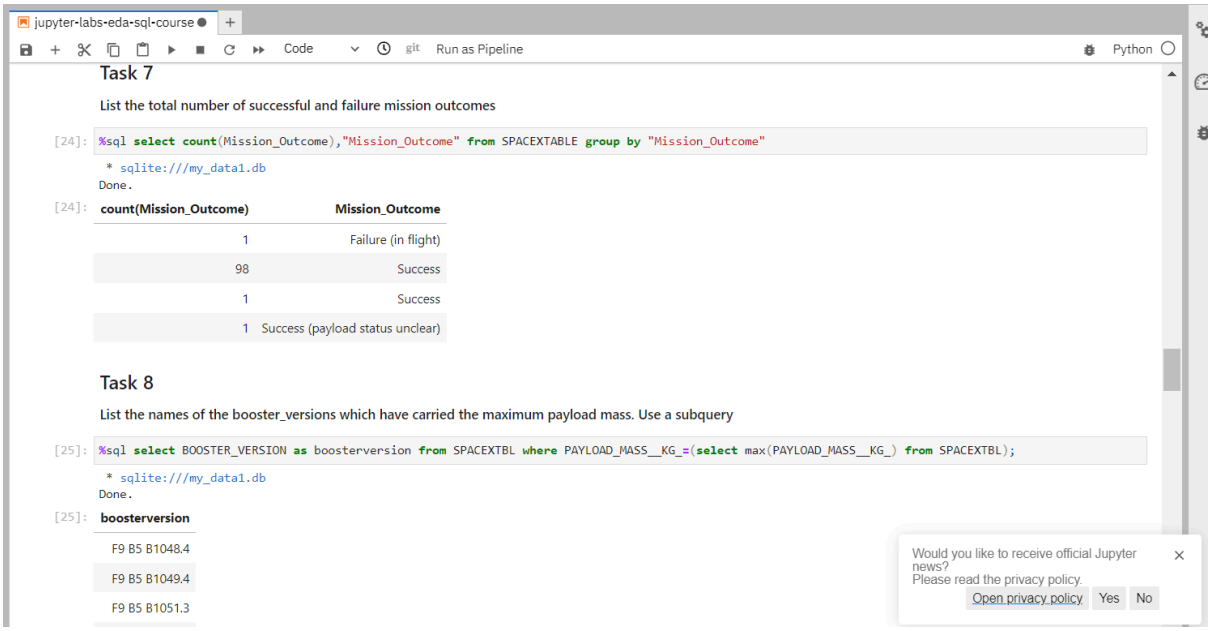

jupyter-labs-eda-sql-course
Code
Run as Pipeline
Python
Spacex DataSet
[2]: !pip install sqlalchemy==1.3.9
Collecting sqlalchemy==1.3.9
Downloading SQLAlchemy-1.3.9.tar.gz (6.0 MB)
6.0/6.0 MB 76.3 MB/s eta 0:00:00:00:0100:01
Preparing metadata (setup.py) ... done
Building wheels for collected packages: sqlalchemy
Building wheel for sqlalchemy (setup.py) ... done
Created wheel for sqlalchemy: filename=SQLAlchemy-1.3.9-cp37-cp37m-linux_x86_64.whl size=1159121 sha256=e0685760a84e21407882a1e45458cd139f7bbff7dcf302f8e8fd516a79e45e85
Stored in directory: /home/jupyterlab/.cache/pip/wheels/03/71/13/010faf12246f72dc76b4150e6e599d13a85b4435e06fb9e51f
Successfully built sqlalchemy
Installing collected packages: sqlalchemy
Attempting uninstall: sqlalchemy
Found existing installation: SQLAlchemy 1.3.24
Uninstalling SQLAlchemy-1.3.24:
Successfully uninstalled SQLAlchemy-1.3.24
Successfully installed sqlalchemy-1.3.9
Connect to the database
Let us first load the SQL extension and establish a connection with the database
[3]: %load_ext sql
[4]: import csv, sqlite3
con = sqlite3.connect("my_data1.db")
cur = con.cursor()
Would you like to receive official Jupyter news?
Please read the privacy policy.
Open privacy policy
Yes
No

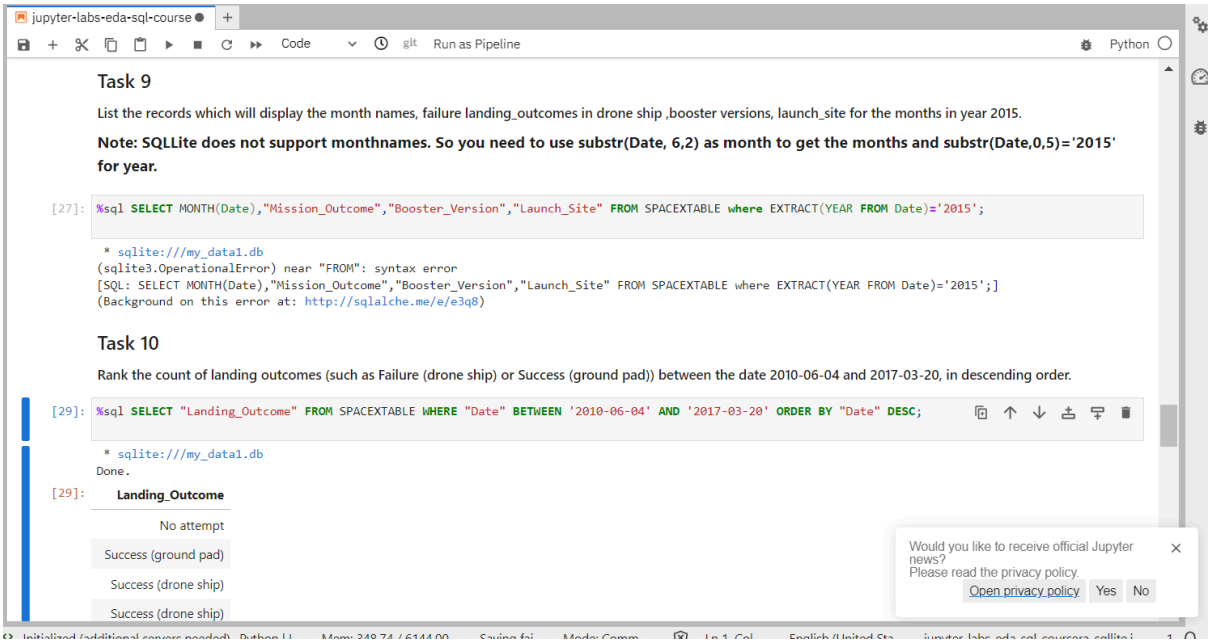

jupyter-labs-eda-sql-course
Code
Run as Pipeline
Python
[3]: %load_ext sql
[4]: import csv, sqlite3
con = sqlite3.connect("my_data1.db")
cur = con.cursor()
[5]: !pip install -q pandas==1.1.5
[6]: %sql sqlite:///my_data1.db
[6]: 'Connected: @my_data1.db'
[7]: import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv")
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/pandas/core/generic.py:2882: UserWarning: The spaces in these column names will not be changed. In pandas versions < 0.14, spaces were converted to underscores.
both result in 0.1234 being formatted as 0.12.
Note:This below code is added to remove blank rows from table
[8]: %sql create table SPACEXTABLE as select * from SPACEXTBL where Date is not null
* sqlite:///my_data1.db
(sqlite3.OperationalError) table SPACEXTABLE already exists
[SQL: create table SPACEXTABLE as select * from SPACEXTBL where Date is not null]
(Background on this error at: http://sqlalche.me/e/e3q8)
Tasks
Would you like to receive official Jupyter news?
Please read the privacy policy.
Open privacy policy
Yes
No


jupyter-labs-eda-sql-course
Code
Run as Pipeline
Python
Tasks
Now write and execute SQL queries to solve the assignment tasks.
Note: If the column names are in mixed case enclose it in double quotes For Example "Landing_Outcome"
Task 1
Display the names of the unique launch sites in the space mission
[9]: %sql SELECT DISTINCT "Launch_Site" FROM SPACEXTABLE
* sqlite:///my_data1.db
Done.
[9]: Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
Task 2
Display 5 records where launch sites begin with the string 'CCA'
[11]: %sql SELECT "Launch_Site" FROM SPACEXTABLE WHERE "Launch_Site" LIKE '%CCA%' LIMIT 5
* sqlite:///my_data1.db
Done.
Would you like to receive official Jupyter news?
Please read the privacy policy.
Open privacy policy
Yes
No
Initialized (additional servers needed)
Python | I...
Mem: 348.74 / 6144.00 ...
Saving fai...
Mode: Comm...
Ln 1, Col ...
English (United Sta...
jupyter-labs-eda-sql-coursera_sqllite.i...


jupyter-labs-eda-sql-course
Code
Run as Pipeline
Python
Task 3
Display the total payload mass carried by boosters launched by NASA (CRS)
[12]: %sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTABLE
* sqlite:///my_data1.db
Done.
[12]: payloadmass
619967
Task 4
Display average payload mass carried by booster version F9 v1.1
[16]: %sql select avg(PAYLOAD_MASS__KG_) as payloadmass_avg from SPACEXTABLE where "Booster_Version" like "F9 v1.1"
* sqlite:///my_data1.db
Done.
[16]: payloadmass_avg
2928.4
Task 5
List the date when the first succesful landing outcome in ground pad was acheived.
Hint:Use min function
Would you like to receive official Jupyter news?
Please read the privacy policy.
Open privacy policy
Yes
No


jupyter-labs-eda-sql-course
Code
Run as Pipeline
Python
Task 5
List the date when the first succesful landing outcome in ground pad was acheived.
Hint:Use min function
[19]: %sql select min(Date) from SPACEXTABLE where "Landing_Outcome" = "Success (ground pad)"
* sqlite:///my_data1.db
Done.
[19]: min(Date)
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
[21]: l select "Booster_Version" from SPACEXTABLE where "Landing_Outcome" = "Success (drone ship)" and "PAYLOAD_MASS__KG_" > 4000 and "PAYLOAD_MASS__KG_" > 6
* sqlite:///my_data1.db
Done.
[21]: Booster_Version
F9 FT B1029.1
F9 FT B1036.1
F9 B4 B1041.1
Would you like to receive official Jupyter news?
Please read the privacy policy.
Open privacy policy
Yes
No


jupyter-labs-eda-sql-course
Code
Run as Pipeline
Python
Task 7
List the total number of successful and failure mission outcomes
[24]: %sql select count(Mission_Outcome),"Mission_Outcome" from SPACEXTABLE group by "Mission_Outcome"
* sqlite:///my_data1.db
Done.
[24]: count(Mission_Outcome) Mission_Outcome
1 Failure (in flight)
98 Success
1 Success
1 Success (payload status unclear)
Task 8
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
[25]: %sql select BOOSTER_VERSION as boosterversion from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
* sqlite:///my_data1.db
Done.
[25]: boosterversion
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
Would you like to receive official Jupyter news?
Please read the privacy policy.
Open privacy policy
Yes
No


jupyter-labs-eda-sql-course
Code
Run as Pipeline
Python
Task 9
List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.
[27]:
%sql SELECT MONTH(Date),"Mission_Outcome","Booster_Version","Launch_Site" FROM SPACEXTABLE where EXTRACT(YEAR FROM Date)='2015';
* sqlite:///my_data1.db
(sqlite3.OperationalError) near "FROM": syntax error
[SQL: SELECT MONTH(Date),"Mission_Outcome","Booster_Version","Launch_Site" FROM SPACEXTABLE where EXTRACT(YEAR FROM Date)='2015';]
(Background on this error at: http://sqlalche.me/e/e3q8)
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.
[29]:
%sql SELECT "Landing_Outcome" FROM SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY "Date" DESC;
* sqlite:///my_data1.db
Done.
[29]:
Landing_Outcome
No attempt
Success (ground pad)
Success (drone ship)
Success (drone ship)
Would you like to receive official Jupyter news?
Please read the privacy policy.
Open privacy policy
Yes
No