COMS W4111: Introduction to Databases Sections 002, V02 Fall 2022

Homework 0 - Environment Setup

Introduction/Overview

Please consult the HW0: Environment PDF for detailed instructions. Complete all the tests in this notebook and submit only this notebook as a PDF to GradeScope. To convert the jupyter notebook into a pdf you can use either of the following methods:

- File --> Print Preview --> Print --> Save to PDF

- File --> Download As HTML --> Print --> Save to PDF

Due date: September 18, 11:59 PM EDT on GradeScope

Please note: You may NOT use late days for the submission of this assignment. Check Courseworks for GradeScope access.

It is recommended that you put the screenshots into the same folder as this notebook so you do not have to alter the path to include your images.

Please read all the instructions thoroughly!

Add Student Information

1. Replace my name with your full name.
2. Replace my UNI with your UNI.
3. Replace "Cool Track" with either "Programming" or "Non-programming."

In [1]:

```
# Print your name, uni, and track below

name = "William Das"
uni = "whd2108"
track = "Programming"

print(name)
print(uni)
print(track)
```

```
William Das
whd2108
Programming
```

Testing Anaconda and Python

Run the following cells to ensure that you have the correct version of Python and all necessary packages installed.

Python Version

If you Python version test failed, you installed Anaconda incorrectly. You will have to uninstall and install a correct, recent version.

In [2]:

```
import sys

print("Python version information:", sys.version_info, "\n")
if sys.version_info.major != 3 or \
    ((sys.version_info.major == 3) and (sys.version_info.minor < 7)):
    print("You have an invalid version of Python.")
else:
    print("Your Python version is OK.")
```

```
Python version information: sys.version_info(major=3, minor=9, micro=12, rel
easelevel='final', serial=0)

Your Python version is OK.
```

Python Path and Information

In [3]:

```
python_found = False
anaconda_found = False

for p in sys.path:
    print(p)
    if "anaconda3" in p:
        print("Found anaconda3")
        anaconda_found = True
    if "python" in p:
        print("Found some kind of Python.")
        if not anaconda_found:
            print("Found some type of Python other than Anaconda.")
            print("Test fails")
        else:
            print("OK. Path is good.")
            python_found = True
        break

if python_found and anaconda_found:
    print("\nPassed all path tests.")
else:
    print("\nFailed path tests.")
```

```
/Users/williamdas/Documents/Intro Databases Fall 2022/F22_W4111_HW_0
/Users/williamdas/opt/anaconda3/lib/python39.zip
Found anaconda3
Found some kind of Python.
OK. Path is good.

Passed all path tests.
```

If you path/environment test failed, you installed Anaconda incorrectly. You will have to uninstall and install a correct, recent version.

Test Conda/Anaconda Version

In [4]:

```
import conda
```

In [5]:

```
conda_version_info =  conda.sys.version_info
print("Your conda version info is\n", conda_version_info)

print("Conda version information:", conda_version_info, "\n")
if conda_version_info.major != 3 or \
    ((conda_version_info.major == 3) and (conda_version_info.minor < 6)):
    print("You have an invalid version of Conda.")
else:
    print("Your Conda version is OK.")
```

```
Your conda version info is
 sys.version_info(major=3, minor=9, micro=12, releaselevel='final', serial=0
)
Conda version information: sys.version_info(major=3, minor=9, micro=12, rele
aselevel='final', serial=0)

Your Conda version is OK.
```

If you the version test failed, you installed Anaconda incorrectly. You will have to uninstall and install a correct, recent version.

Test Pandas

In [6]:

```python
import pandas
p_version = pandas.__version__
p_nums = p_version.split(".")

print("Your pandas version is ", p_version)
if p_nums[0] != '1':
    print("Your version is invalid.")
else:
    print("Your version is OK.")

# This checks to see if you are on pandas 1.0.5 or 1.2.0 both of which are C
```

```
Your pandas version is  1.4.2
Your version is OK.
```

If you do not have Pandas already you will need to install Pandas using the following cell:

In [7]:

```
!pip install pandas
```

```
Requirement already satisfied: pandas in /Users/williamdas/opt/anaconda3/lib
/python3.9/site-packages (1.4.2)
Requirement already satisfied: python-dateutil>=2.8.1 in /Users/williamdas/o
pt/anaconda3/lib/python3.9/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /Users/williamdas/opt/anacond
a3/lib/python3.9/site-packages (from pandas) (2021.3)
Requirement already satisfied: numpy>=1.18.5 in /Users/williamdas/opt/anacon
da3/lib/python3.9/site-packages (from pandas) (1.21.5)
Requirement already satisfied: six>=1.5 in /Users/williamdas/opt/anaconda3/l
ib/python3.9/site-packages (from python-dateutil>=2.8.1->pandas) (1.16.0)
```

Install ipython-sql

In [8]:

```
!pip install ipython-sql
```

```
Collecting ipython-sql
  Downloading ipython_sql-0.4.1-py3-none-any.whl (21 kB)
Requirement already satisfied: sqlalchemy>=0.6.7 in /Users/williamdas/opt/an
```

```
aconda3/lib/python3.9/site-packages (from ipython-sql) (1.4.32)
Collecting prettytable<1
  Downloading prettytable-0.7.2.zip (28 kB)
Requirement already satisfied: six in /Users/williamdas/opt/anaconda3/lib/py
thon3.9/site-packages (from ipython-sql) (1.16.0)
Requirement already satisfied: ipython-genutils>=0.1.0 in /Users/williamdas/
opt/anaconda3/lib/python3.9/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: ipython>=1.0 in /Users/williamdas/opt/anacond
a3/lib/python3.9/site-packages (from ipython-sql) (8.2.0)
Collecting sqlparse
  Using cached sqlparse-0.4.2-py3-none-any.whl (42 kB)
Requirement already satisfied: appnope in /Users/williamdas/opt/anaconda3/li
b/python3.9/site-packages (from ipython>=1.0->ipython-sql) (0.1.2)
Requirement already satisfied: pexpect>4.3 in /Users/williamdas/opt/anaconda
3/lib/python3.9/site-packages (from ipython>=1.0->ipython-sql) (4.8.0)
Requirement already satisfied: decorator in /Users/williamdas/opt/anaconda3/
lib/python3.9/site-packages (from ipython>=1.0->ipython-sql) (5.1.1)
Requirement already satisfied: pygments>=2.4.0 in /Users/williamdas/opt/anac
onda3/lib/python3.9/site-packages (from ipython>=1.0->ipython-sql) (2.11.2)
Requirement already satisfied: prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0
in /Users/williamdas/opt/anaconda3/lib/python3.9/site-packages (from ipython
>=1.0->ipython-sql) (3.0.20)
Requirement already satisfied: traitlets>=5 in /Users/williamdas/opt/anacond
a3/lib/python3.9/site-packages (from ipython>=1.0->ipython-sql) (5.1.1)
Requirement already satisfied: matplotlib-inline in /Users/williamdas/opt/an
aconda3/lib/python3.9/site-packages (from ipython>=1.0->ipython-sql) (0.1.2)
Requirement already satisfied: stack-data in /Users/williamdas/opt/anaconda3
/lib/python3.9/site-packages (from ipython>=1.0->ipython-sql) (0.2.0)
Requirement already satisfied: jedi>=0.16 in /Users/williamdas/opt/anaconda3
/lib/python3.9/site-packages (from ipython>=1.0->ipython-sql) (0.18.1)
Requirement already satisfied: pickleshare in /Users/williamdas/opt/anaconda
3/lib/python3.9/site-packages (from ipython>=1.0->ipython-sql) (0.7.5)
Requirement already satisfied: backcall in /Users/williamdas/opt/anaconda3/l
ib/python3.9/site-packages (from ipython>=1.0->ipython-sql) (0.2.0)
Requirement already satisfied: setuptools>=18.5 in /Users/williamdas/opt/ana
conda3/lib/python3.9/site-packages (from ipython>=1.0->ipython-sql) (61.2.0)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /Users/williamdas/opt/
anaconda3/lib/python3.9/site-packages (from jedi>=0.16->ipython>=1.0->ipytho
n-sql) (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in /Users/williamdas/opt/anac
onda3/lib/python3.9/site-packages (from pexpect>4.3->ipython>=1.0->ipython-s
ql) (0.7.0)
Requirement already satisfied: wcwidth in /Users/williamdas/opt/anaconda3/li
b/python3.9/site-packages (from prompt-toolkit!=3.0.0,!=3.0.1,<3.1.0,>=2.0.0
->ipython>=1.0->ipython-sql) (0.2.5)
Requirement already satisfied: greenlet!=0.4.17 in /Users/williamdas/opt/ana
conda3/lib/python3.9/site-packages (from sqlalchemy>=0.6.7->ipython-sql) (1.
1.1)
Requirement already satisfied: pure-eval in /Users/williamdas/opt/anaconda3/
lib/python3.9/site-packages (from stack-data->ipython>=1.0->ipython-sql) (0.
2.2)
Requirement already satisfied: executing in /Users/williamdas/opt/anaconda3/
lib/python3.9/site-packages (from stack-data->ipython>=1.0->ipython-sql) (0.
8.3)
Requirement already satisfied: asttokens in /Users/williamdas/opt/anaconda3/
lib/python3.9/site-packages (from stack-data->ipython>=1.0->ipython-sql) (2.
0.5)
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=a54cfb7ceccaa19e4f8aea69a41850617d1c2850dcb45d34ef59480038
ea9b2f
  Stored in directory: /Users/williamdas/Library/Caches/pip/wheels/75/f7/28/
77a076f1fa8cbeda61aca712815d04d7a32435f04a26a2dd7b
Successfully built prettytable
Installing collected packages: sqlparse, prettytable, ipython-sql
Successfully installed ipython-sql-0.4.1 prettytable-0.7.2 sqlparse-0.4.2
```

- If you got errors, please follow the instructions in the ipython-sql site to install the magic.
- **NOTE:** Running the cell above may produce multiple notifications about installing requirements or requirement already satisfied. That is normal.
- Once you get the install to work without errors, run the following cell.

In [17]:

```
%load_ext sql
```

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

- If you did not get an error response, your test passed.
- If you run the cell twice, your answer should be:

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

SQLAlchemy/PyMySQL

Install `sqlalchemy` and `pymysql.` These are Python language packages for interacting with SQL and MySQL databases.

In [10]:

```
!pip install sqlalchemy
!pip install pymysql
```

```
Requirement already satisfied: sqlalchemy in /Users/williamdas/opt/anaconda3
/lib/python3.9/site-packages (1.4.32)
Requirement already satisfied: greenlet!=0.4.17 in /Users/williamdas/opt/ana
conda3/lib/python3.9/site-packages (from sqlalchemy) (1.1.1)
Collecting pymysql
  Downloading PyMySQL-1.0.2-py3-none-any.whl (43 kB)
     |████████████████████████████████| 43 kB 3.8 MB/s eta 0:00:011
Installing collected packages: pymysql
Successfully installed pymysql-1.0.2
```

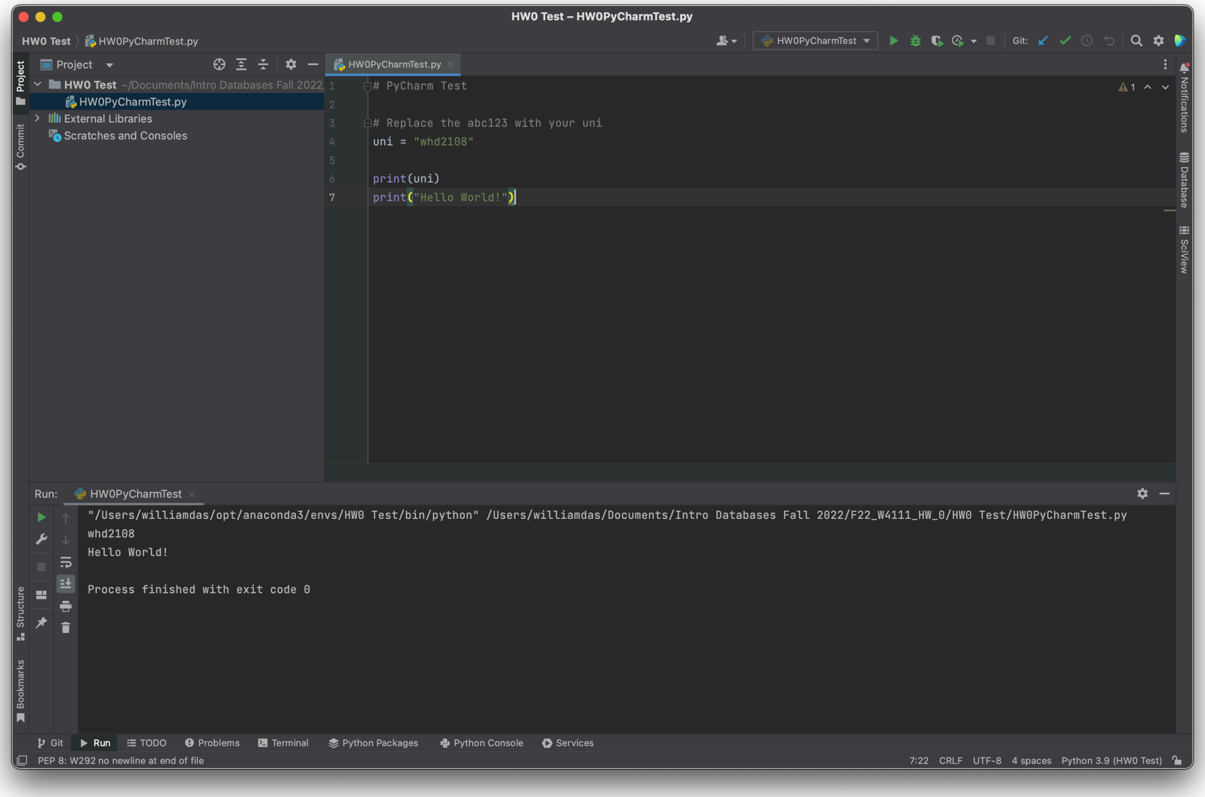
PyCharm

Required for Programming Track only, but recommended for all. Follow the instructions to setup PyCharm and run the test. Take a screenshot and insert it into the notebook using the cell below. You may have to change the path to the name and/or location of your image.

In [12]:

```
from IPython.display import Image

Image("./PyCharmScreenshot.png")
```

Out[12]:



```
# PyCharm Test

# Replace the abc123 with your uni
uni = "whd2108"

print(uni)
print("Hello World!")
```

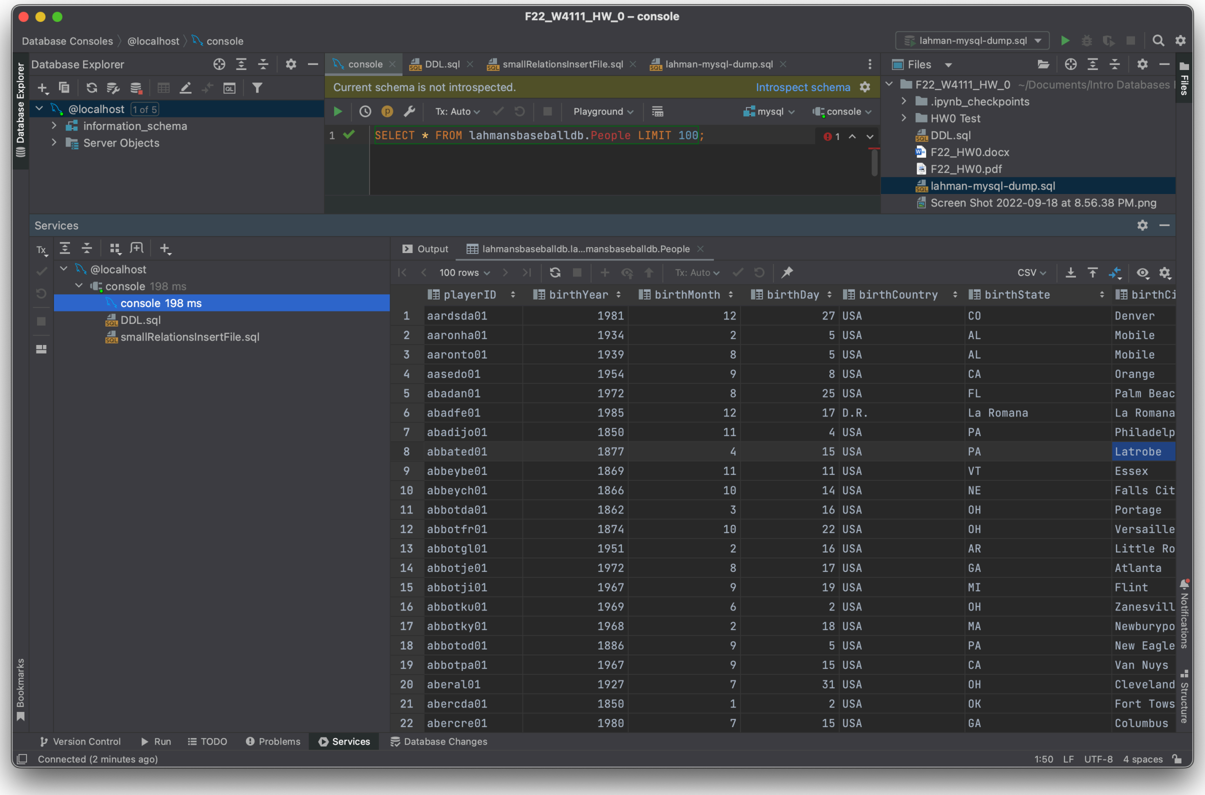
```
"/Users/williamdas/opt/anaconda3/envs/HW0 Test/bin/python" /Users/williamdas/Documents/Intro Databases Fall 2022/F22_W4111_HW_0/HW0 Test/HW0PyCharmTest.py
whd2108
Hello World!

Process finished with exit code 0
```

DataGrip

Follow the instructions to setup DataGrip and connect DataGrip to your AWS server. Insert your screenshot of the successful query on the Lahman database into the notebook using the cell below. You may have to change the path to the name and/or location of your image.

In [13]:
```
Image("./DataGripScreenshot.png")
```

Out[13]:



Current schema is not introspected. Introspect schema

```
SELECT * FROM lahmansbaseballdb.People LIMIT 100;
```

Connected (2 minutes ago)

	playerID	birthYear	birthMonth	birthDay	birthCountry	birthState	birthCi
1	aardsda01	1981	12	27	USA	CO	Denver
2	aaronha01	1934	2	5	USA	AL	Mobile
3	aaronto01	1939	8	5	USA	AL	Mobile
4	aasedo01	1954	9	8	USA	CA	Orange
5	abadan01	1972	8	25	USA	FL	Palm Beac
6	abadfe01	1985	12	17	D.R.	La Romana	La Romana
7	abadijo01	1850	11	4	USA	PA	Philadelp
8	abbated01	1877	4	15	USA	PA	Latrobe
9	abbeybe01	1869	11	11	USA	VT	Essex
10	abbeych01	1866	10	14	USA	NE	Falls Cit
11	abbotda01	1862	3	16	USA	OH	Portage
12	abbotfr01	1874	10	22	USA	OH	Versaille
13	abbotgl01	1951	2	16	USA	AR	Little Ro
14	abbotje01	1972	8	17	USA	GA	Atlanta
15	abbotji01	1967	9	19	USA	MI	Flint
16	abbotku01	1969	6	2	USA	OH	Zanesvill
17	abbotky01	1968	2	18	USA	MA	Newburypo
18	abbotod01	1886	9	5	USA	PA	New Eagle
19	abbotpa01	1967	9	15	USA	CA	Van Nuys
20	aberal01	1927	7	31	USA	OH	Cleveland
21	abercda01	1850	1	2	USA	OK	Fort Tows
22	abercre01	1980	7	15	USA	GA	Columbus

The code below indicates how to connect this notebook to your AWS Database.

You will need to change the username, password, and endpoint to match

In [15]:
```
%load_ext sql
%sql mysql+pymysql://root:dbuserbdbuser@localhost:3306/lahmansbaseballdb
```

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

Run the cell below to query the AWS database from the notebook:

In [16]:
```
%sql SELECT * FROM lahmansbaseballdb.People LIMIT 10;
```

```
 * mysql+pymysql://root:***@localhost:3306/lahmansbaseballdb
10 rows affected.
```

Out[16]:

playerID	birthYear	birthMonth	birthDay	birthCountry	birthState	birthCity	deathYear
aardsda01	1981	12	27	USA	CO	Denver	None
aaronha01	1934	2	5	USA	AL	Mobile	None
aaronto01	1939	8	5	USA	AL	Mobile	1984
aasedo01	1954	9	8	USA	CA	Orange	None
abadan01	1972	8	25	USA	FL	Palm Beach	None
abadfe01	1985	12	17	D.R.	La Romana	La Romana	None
abadijo01	1850	11	4	USA	PA	Philadelphia	1905
abbated01	1877	4	15	USA	PA	Latrobe	1957
abbeybe01	1869	11	11	USA	VT	Essex	1962
abbeych01	1866	10	14	USA	NE	Falls City	1926

In [ ]: