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 = "Honghao Liu"
uni = "hl3630"
track = "Programming"

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

```
Honghao Liu
hl3630
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=8, micro=13, releaseleve
l='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.")
```

```
C:\Users\16466\OneDrive\Documents\CU\4111 intro to DB\F22_W4111_HW_0\F22_W4111_HW_0
C:\Users\16466\.conda\envs\DB\python38.zip
Found some kind of Python.
Found some type of Python other than Anaconda.
Test fails

Failed 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 [ ]:

```
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, releaselevel
='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 [7]:

```
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 OK
```

```
Your pandas version is  1.4.4
Your version is OK.
```

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

In [5]:

```
!pip install pandas
```

```
Collecting pandas
  Downloading pandas-1.4.4-cp38-cp38-win_amd64.whl (10.6 MB)
     ---------------------------------------- 10.6/10.6 MB 7.4 MB/s eta 0:00:00
Collecting pytz>=2020.1
  Downloading pytz-2022.2.1-py2.py3-none-any.whl (500 kB)
     --------------------------------------- 500.6/500.6 kB 7.8 MB/s eta 0:00:00
Collecting numpy>=1.18.5
  Downloading numpy-1.23.3-cp38-cp38-win_amd64.whl (14.7 MB)
     ---------------------------------------- 14.7/14.7 MB 7.6 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\16466\.conda\envs
\db\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\16466\.conda\envs\db\lib\site-pa
ckages (from python-dateutil>=2.8.1->pandas) (1.16.0)
Installing collected packages: pytz, numpy, pandas
Successfully installed numpy-1.23.3 pandas-1.4.4 pytz-2022.2.1
```

Install ipython-sql

In [6]:

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

```
Collecting ipython-sql
  Using cached ipython_sql-0.4.1-py3-none-any.whl (21 kB)
Requirement already satisfied: six in c:\users\16466\.conda\envs\db\lib\site-package
s (from ipython-sql) (1.16.0)
Collecting sqlparse
  Using cached sqlparse-0.4.2-py3-none-any.whl (42 kB)
Collecting prettytable<1
  Using cached prettytable-0.7.2.zip (28 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Requirement already satisfied: ipython>=1.0 in c:\users\16466\.conda\envs\db\lib\sit
e-packages (from ipython-sql) (8.5.0)
Requirement already satisfied: ipython-genutils>=0.1.0 in c:\users\16466\.conda\envs
\db\lib\site-packages (from ipython-sql) (0.2.0)
Collecting sqlalchemy>=0.6.7
  Downloading SQLAlchemy-1.4.41-cp38-cp38-win_amd64.whl (1.6 MB)
     ---------------------------------------- 1.6/1.6 MB 2.0 MB/s eta 0:00:00
Requirement already satisfied: pickleshare in c:\users\16466\.conda\envs\db\lib\site
-packages (from ipython>=1.0->ipython-sql) (0.7.5)
Requirement already satisfied: backcall in c:\users\16466\.conda\envs\db\lib\site-pa
ckages (from ipython>=1.0->ipython-sql) (0.2.0)
Requirement already satisfied: jedi>=0.16 in c:\users\16466\.conda\envs\db\lib\site-
packages (from ipython>=1.0->ipython-sql) (0.18.1)
Requirement already satisfied: stack-data in c:\users\16466\.conda\envs\db\lib\site-
packages (from ipython>=1.0->ipython-sql) (0.5.0)
Requirement already satisfied: colorama in c:\users\16466\.conda\envs\db\lib\site-pa
ckages (from ipython>=1.0->ipython-sql) (0.4.5)
Requirement already satisfied: traitlets>=5 in c:\users\16466\.conda\envs\db\lib\sit
e-packages (from ipython>=1.0->ipython-sql) (5.4.0)
Requirement already satisfied: pygments>=2.4.0 in c:\users\16466\.conda\envs\db\lib
\site-packages (from ipython>=1.0->ipython-sql) (2.13.0)
Requirement already satisfied: decorator in c:\users\16466\.conda\envs\db\lib\site-p
ackages (from ipython>=1.0->ipython-sql) (5.1.1)
Requirement already satisfied: matplotlib-inline in c:\users\16466\.conda\envs\db\li
b\site-packages (from ipython>=1.0->ipython-sql) (0.1.6)
Requirement already satisfied: prompt-toolkit<3.1.0,>3.0.1 in c:\users\16466\.conda
\envs\db\lib\site-packages (from ipython>=1.0->ipython-sql) (3.0.31)
Collecting greenlet!=0.4.17
  Downloading greenlet-1.1.3-cp38-cp38-win_amd64.whl (101 kB)
     -------------------------------------- 101.5/101.5 kB 1.2 MB/s eta 0:00:00
Requirement already satisfied: parso<0.9.0,>=0.8.0 in c:\users\16466\.conda\envs\db
\lib\site-packages (from jedi>=0.16->ipython>=1.0->ipython-sql) (0.8.3)
Requirement already satisfied: wcwidth in c:\users\16466\.conda\envs\db\lib\site-pac
kages (from prompt-toolkit<3.1.0,>3.0.1->ipython>=1.0->ipython-sql) (0.2.5)
Requirement already satisfied: asttokens in c:\users\16466\.conda\envs\db\lib\site-p
ackages (from stack-data->ipython>=1.0->ipython-sql) (2.0.8)
Requirement already satisfied: pure-eval in c:\users\16466\.conda\envs\db\lib\site-p
ackages (from stack-data->ipython>=1.0->ipython-sql) (0.2.2)
Requirement already satisfied: executing in c:\users\16466\.conda\envs\db\lib\site-p
ackages (from stack-data->ipython>=1.0->ipython-sql) (1.0.0)
Building wheels for collected packages: prettytable
  Building wheel for prettytable (setup.py): started
  Building wheel for prettytable (setup.py): finished with status 'done'
  Created wheel for prettytable: filename=prettytable-0.7.2-py3-none-any.whl size=13
695 sha256=f45c311c1858384cd83fd2251782eb1d7e7c7f8eb60230588140b13ace3691a0
  Stored in directory: c:\users\16466\appdata\local\pip\cache\wheels\48\6d\77\9517cb
933af254f51a446f1a5ec9c2be3e45f17384940bce68
Successfully built prettytable
Installing collected packages: prettytable, sqlparse, greenlet, sqlalchemy, ipython-
sql
Successfully installed greenlet-1.1.3 ipython-sql-0.4.1 prettytable-0.7.2 sqlalchemy
-1.4.41 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 [9]:

```
%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 c:\users\16466\.conda\envs\db\lib\site-
packages (1.4.41)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\16466\.conda\envs\db\lib
\site-packages (from sqlalchemy) (1.1.3)
Collecting pymysql
  Downloading PyMySQL-1.0.2-py3-none-any.whl (43 kB)
     ---------------------------------------- 43.8/43.8 kB 2.1 MB/s eta 0:00:00
Installing collected packages: pymysql
Successfully installed pymysql-1.0.2
```

PyCharm

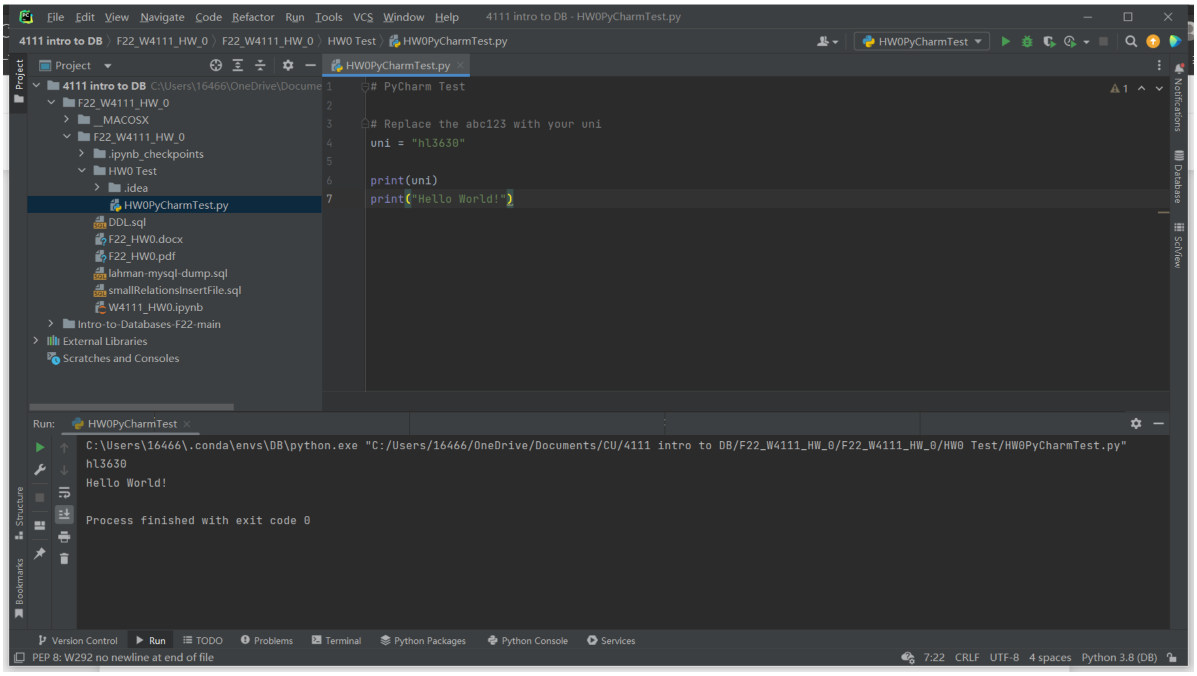
Required for Programming Track only, but recommended for all. Follow the instructions to setup PyCharm and run the test. Take a screenshot and insert it into the notebook using the cell below. You may have to change the path to the name and/or location of your image.

In [18]:

```
from IPython.display import Image


Image('PyCharmScreenshot.png')
```

Out[18]:



File Edit View Navigate Code Refactor Run Tools VCS Window Help
4111 intro to DB - HW0PyCharmTest.py
4111 intro to DB 〉 F22_W4111_HW_0 〉 F22_W4111_HW_0 〉 HW0 Test 〉 HW0PyCharmTest.py
HW0PyCharmTest
Project
4111 intro to DB C:\Users\16466\OneDrive\Docume
F22_W4111_HW_0
__MACOSX
F22_W4111_HW_0
.ipynb_checkpoints
HW0 Test
.idea
HW0PyCharmTest.py
DDL.sql
F22_HW0.docx
F22_HW0.pdf
lahman-mysql-dump.sql
smallRelationsInsertFile.sql
W4111_HW0.ipynb
Intro-to-Databases-F22-main
External Libraries
Scratches and Consoles
HW0PyCharmTest.py
Run: HW0PyCharmTest
Version Control Run TODO Problems Terminal Python Packages Python Console Services
PEP 8: W292 no newline at end of file
7:22 CRLF UTF-8 4 spaces Python 3.8 (DB)

```
# PyCharm Test

# Replace the abc123 with your uni
uni = "hl3630"

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

```
C:\Users\16466\.conda\envs\DB\python.exe "C:/Users/16466/OneDrive/Documents/CU/4111 intro to DB/F22_W4111_HW_0/F22_W4111_HW_0/HW0 Test/HW0PyCharmTest.py"
hl3630
Hello World!

Process finished with exit code 0
```

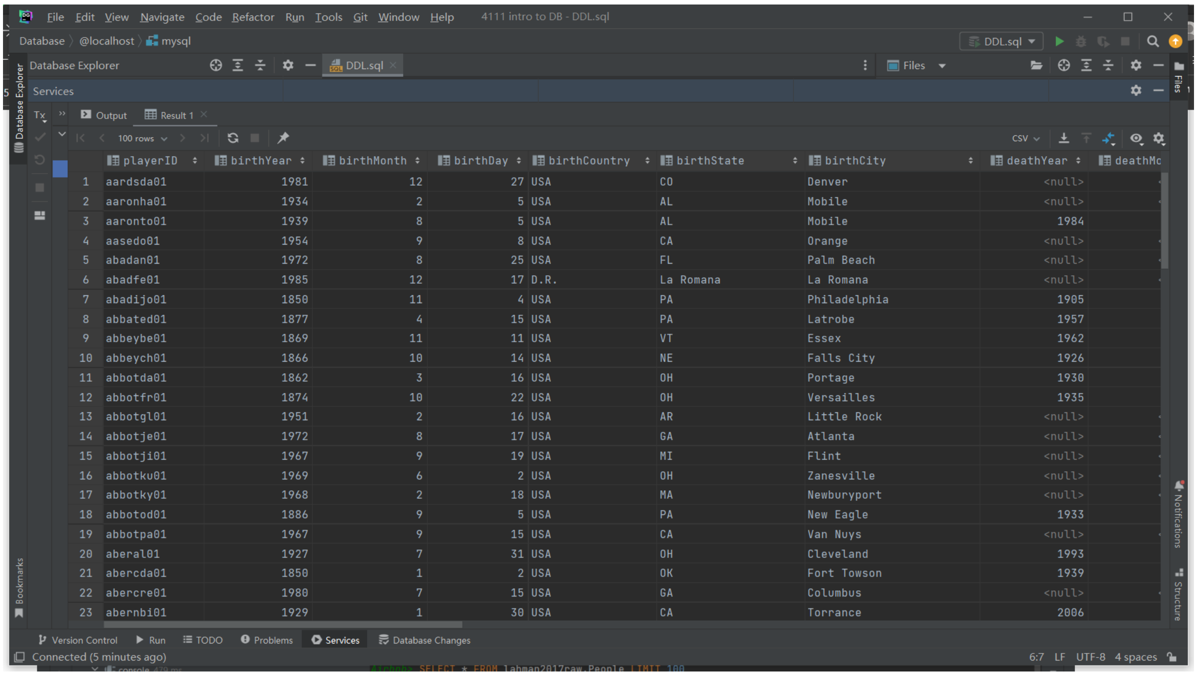
DataGrip

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 [19]:

```
Image("DataGripScreenshot.png")
```

Out[19]:



File Edit View Navigate Code Refactor Run Tools Git Window Help
4111 intro to DB - DDL.sql
Database 〉 @localhost 〉 mysql
DDL.sql
Database Explorer
Files
Services
Tx
Output
Result 1
100 rows
CSV
Version Control Run TODO Problems Services Database Changes
Connected (5 minutes ago)
6:7 LF UTF-8 4 spaces

	playerID	birthYear	birthMonth	birthDay	birthCountry	birthState	birthCity	deathYear	deathMc
1	aardsda01	1981	12	27	USA	CO	Denver	<null>	
2	aaronha01	1934	2	5	USA	AL	Mobile	<null>	
3	aaronto01	1939	8	5	USA	AL	Mobile	1984	
4	aasedo01	1954	9	8	USA	CA	Orange	<null>	
5	abadan01	1972	8	25	USA	FL	Palm Beach	<null>	
6	abadfe01	1985	12	17	D.R.	La Romana	La Romana	<null>	
7	abadijo01	1850	11	4	USA	PA	Philadelphia	1905	
8	abbated01	1877	4	15	USA	PA	Latrobe	1957	
9	abbeybe01	1869	11	11	USA	VT	Essex	1962	
10	abbeych01	1866	10	14	USA	NE	Falls City	1926	
11	abbotda01	1862	3	16	USA	OH	Portage	1930	
12	abbotfr01	1874	10	22	USA	OH	Versailles	1935	
13	abbotgl01	1951	2	16	USA	AR	Little Rock	<null>	
14	abbotje01	1972	8	17	USA	GA	Atlanta	<null>	
15	abbotji01	1967	9	19	USA	MI	Flint	<null>	
16	abbotku01	1969	6	2	USA	OH	Zanesville	<null>	
17	abbotky01	1968	2	18	USA	MA	Newburyport	<null>	
18	abbotod01	1886	9	5	USA	PA	New Eagle	1933	
19	abbotpa01	1967	9	15	USA	CA	Van Nuys	<null>	
20	aberal01	1927	7	31	USA	OH	Cleveland	1993	
21	abercda01	1850	1	2	USA	OK	Fort Towson	1939	
22	abercre01	1980	7	15	USA	GA	Columbus	<null>	
23	abernbi01	1929	1	30	USA	CA	Torrance	2006	

```
SELECT * FROM lahman2017raw.People LIMIT 100
```

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 [13]:

```
%load_ext sql
%sql mysql+pymysql://root:dbuserbdbuser@localhost/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 [14]: %sql SELECT * FROM lahmansbaseballdb.People LIMIT 10;
```

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

Out[14]:

playerID	birthYear	birthMonth	birthDay	birthCountry	birthState	birthCity	deathYear	de
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 [ ]:
```