COMS W4111: Introduction to Databases Spring 2024, Sections 002/V02

Homework 0: Environment Setup and Test

Introduction and Overview

HW Objectives

This section of *W4111 - Intro. to Databases* defines required and recommended SW tools. Students are busy and often defer tasks to "just before the homework is due." In previous semesters, students were struggling with SW installation right before the HW 1 submission deadline. To avoid this problem, HW 0 tests the installation and configuration of the SW.

All students must complete and submit HW 0. **You may not use late days.** If you do not submit HW 0 by the deadline, we will deduct 3 points from your HW 1 score.

We will create an entry on GradeScope for HW 0 submission.

We will also create a topic/post on Ed for discussing the submission.

Submission 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: 28-January, 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

In the cell below,

1. Replace `"Your Name"` your full name.
2. Replace `"UNI"` with your UNI.
3. Replace `"Cool Track"` with either `"Programming"` or `"Non-programming,"` depending on the track you have chosen.

Run the cell.

In [1]:
```
# Print your name, uni, and track below

name = "Jiacheng Gu"
uni = "jg4874"
track = "Programming"

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

```
Jiacheng Gu
jg4874
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

The test below should return the path to the Python interpreter for your Anaconda environment. The exact path may be differ from Mac to Windows, or based on installation choices you made. As long as the path has a sufficient level "anaconda3" in it, you should be OK.

The code cells below have the results of my execution. You must execute on your computer and show your results.

Your results will be similar but different in details because you are on a different computer.

In [1]:
```
#
# Run this cell to print your current working directory.
%pwd
```

Out[1]:
```
'/Users/gujiacheng/Columbia/Introuce to Databases/docs/Homework/HW0'
```

In [2]:
```
#
# Display your node information.
import socket
hostname = socket.gethostname()
print("Your values")
print("Host name = ", hostname)
IPAddr = socket.gethostbyname(hostname)
print("IPAddr = ", IPAddr)
```

```
Your values
Host name =  gujiachengdeMBP
IPAddr =  192.168.1.155
```

In [3]:
```
import sys
```

In [4]:
```
ex = sys.executable
ex
```

Out[4]:
```
'/opt/anaconda3/bin/python'
```

In [5]:
```
# Checking that anacoda3 is in the path.
#
if 'anaconda3' in ex:
    print("Test seems OK.")
else:
    print("Not cool.")
```

```
Test seems OK.
```

The following tests that you have a sufficiently up to date version of Python.

In [6]:
```
print("Python version information:\n","\t", sys.version_info, "\n")
if sys.version_info.major != 3 or \
    ((sys.version_info.major == 3) and (sys.version_info.minor < 9)):
    print("You have an invalid version of Python.")
else:
    print("Your Python version is OK.")
```

```
Python version information:
         sys.version_info(major=3, minor=12, micro=4, releaselevel='final',
serial=0)

Your Python version is OK.
```

If the test fails, you have to install Anaconda properly.

Install ipython-sql

The actual message below will vary based on what you do/do not already have installed. You are fine as long as there is not a major error.

In [7]:
```
%pip install ipython-sql
```

```
Collecting ipython-sql
  Downloading ipython_sql-0.5.0-py3-none-any.whl.metadata (17 kB)
Collecting prettytable (from ipython-sql)
  Downloading prettytable-3.11.0-py3-none-any.whl.metadata (30 kB)
Requirement already satisfied: ipython in /opt/anaconda3/lib/python3.12/site
-packages (from ipython-sql) (8.25.0)
Requirement already satisfied: sqlalchemy>=2.0 in /opt/anaconda3/lib/python
3.12/site-packages (from ipython-sql) (2.0.30)
Requirement already satisfied: sqlparse in /opt/anaconda3/lib/python3.12/sit
e-packages (from ipython-sql) (0.5.1)
Requirement already satisfied: six in /opt/anaconda3/lib/python3.12/site-pac
kages (from ipython-sql) (1.16.0)
Requirement already satisfied: ipython-genutils in /opt/anaconda3/lib/python
3.12/site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: typing-extensions>=4.6.0 in /opt/anaconda3/li
b/python3.12/site-packages (from sqlalchemy>=2.0->ipython-sql) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in /opt/anaconda3/lib/python
3.12/site-packages (from sqlalchemy>=2.0->ipython-sql) (3.0.1)
Requirement already satisfied: decorator in /opt/anaconda3/lib/python3.12/si
te-packages (from ipython->ipython-sql) (5.1.1)
Requirement already satisfied: jedi>=0.16 in /opt/anaconda3/lib/python3.12/s
ite-packages (from ipython->ipython-sql) (0.18.1)
Requirement already satisfied: matplotlib-inline in /opt/anaconda3/lib/pytho
n3.12/site-packages (from ipython->ipython-sql) (0.1.6)
Requirement already satisfied: prompt-toolkit<3.1.0,>=3.0.41 in /opt/anacond
a3/lib/python3.12/site-packages (from ipython->ipython-sql) (3.0.43)
Requirement already satisfied: pygments>=2.4.0 in /opt/anaconda3/lib/python
3.12/site-packages (from ipython->ipython-sql) (2.15.1)
Requirement already satisfied: stack-data in /opt/anaconda3/lib/python3.12/s
ite-packages (from ipython->ipython-sql) (0.2.0)
Requirement already satisfied: traitlets>=5.13.0 in /opt/anaconda3/lib/pytho
n3.12/site-packages (from ipython->ipython-sql) (5.14.3)
Requirement already satisfied: pexpect>4.3 in /opt/anaconda3/lib/python3.12/
site-packages (from ipython->ipython-sql) (4.8.0)
Requirement already satisfied: wcwidth in /opt/anaconda3/lib/python3.12/site
-packages (from prettytable->ipython-sql) (0.2.5)
Requirement already satisfied: parso<0.9.0,>=0.8.0 in /opt/anaconda3/lib/pyt
hon3.12/site-packages (from jedi>=0.16->ipython->ipython-sql) (0.8.3)
Requirement already satisfied: ptyprocess>=0.5 in /opt/anaconda3/lib/python
3.12/site-packages (from pexpect>4.3->ipython->ipython-sql) (0.7.0)
Requirement already satisfied: executing in /opt/anaconda3/lib/python3.12/si
te-packages (from stack-data->ipython->ipython-sql) (0.8.3)
Requirement already satisfied: asttokens in /opt/anaconda3/lib/python3.12/si
te-packages (from stack-data->ipython->ipython-sql) (2.0.5)
Requirement already satisfied: pure-eval in /opt/anaconda3/lib/python3.12/si
te-packages (from stack-data->ipython->ipython-sql) (0.2.2)
Downloading ipython_sql-0.5.0-py3-none-any.whl (20 kB)
Downloading prettytable-3.11.0-py3-none-any.whl (28 kB)
Installing collected packages: prettytable, ipython-sql
Successfully installed ipython-sql-0.5.0 prettytable-3.11.0
Note: you may need to restart the kernel to use updated packages.
```

- 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 [3]:
```
%load_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. Your actual response message may be different. Your environment is OK if you do not get a major error.

In [10]:
```
%pip install sqlalchemy
%pip install pymysql
```

```
Requirement already satisfied: sqlalchemy in /opt/anaconda3/lib/python3.12/s
ite-packages (2.0.30)
Requirement already satisfied: typing-extensions>=4.6.0 in /opt/anaconda3/li
b/python3.12/site-packages (from sqlalchemy) (4.11.0)
Requirement already satisfied: greenlet!=0.4.17 in /opt/anaconda3/lib/python
3.12/site-packages (from sqlalchemy) (3.0.1)
Note: you may need to restart the kernel to use updated packages.
Collecting pymysql
  Using cached PyMySQL-1.1.1-py3-none-any.whl.metadata (4.4 kB)
Using cached PyMySQL-1.1.1-py3-none-any.whl (44 kB)
Installing collected packages: pymysql
Successfully installed pymysql-1.1.1
Note: you may need to restart the kernel to use updated packages.
```

MySQL Connectivity

You installed MySQL Community Edition. You have to choose a userID and password during the installation.

Please set the values in the cell below.

In [4]:
```
#
# Replace root with the user ID for MySQL and dbuserdbuser with the passwor
#
%sql mysql+pymysql://root:Gjc20021013!@localhost
```

In [5]:
```
#
# Your list of databases will be different.
#
%sql show databases;
```

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

Out[5]:

Database
information_schema
mysql
p1_database
performance_schema
sys

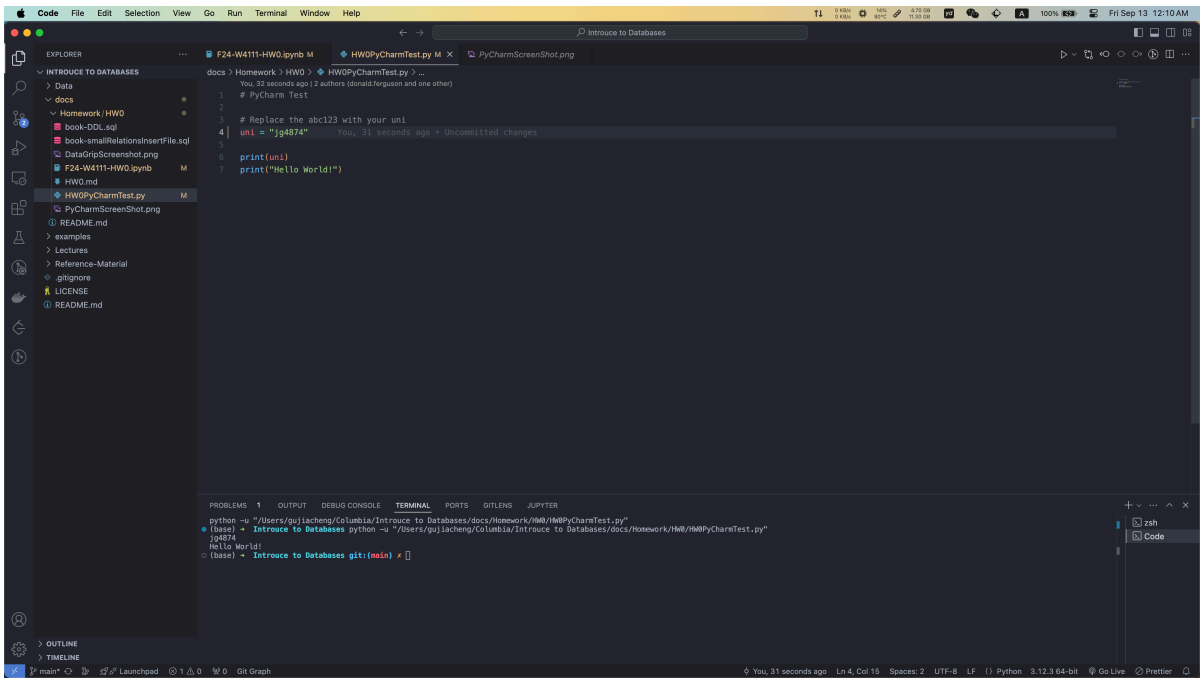
PyCharm

Required for Programming Track only, but recommended for all. Follow the instructions to setup PyCharm and launch. 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 [6]:
```
from IPython.display import Image

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

Out[6]:

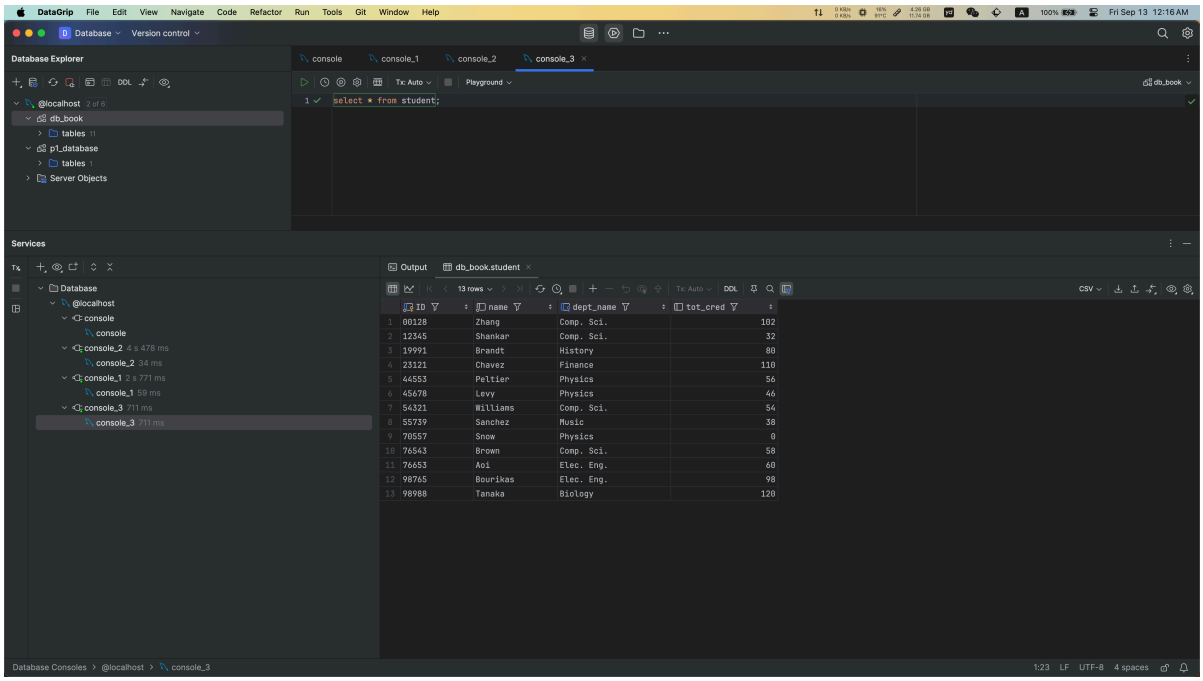


DataGrip

Follow the instructions in the homework definition to setup DataGrip and connect DataGrip to MySQL. Insert your screenshot of the successful query on the sample database below. You may have to change the path to the name and/or location of your image.

In [7]:
```
Image("./DataGripScreenshot.png")
```

Out[7]:



Sample Database

The recitation showed how to install the first database/dataset we will use in the course.

Please follow the recitation and load the data, then run the query below.

In [8]:
```
%sql select * from db_book.student
```

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

Out[8]:

ID	name	dept_name	tot_cred
00128	Zhang	Comp. Sci.	102
12345	Shankar	Comp. Sci.	32
19991	Brandt	History	80
23121	Chavez	Finance	110
44553	Peltier	Physics	56
45678	Levy	Physics	46
54321	Williams	Comp. Sci.	54
55739	Sanchez	Music	38
70557	Snow	Physics	0
76543	Brown	Comp. Sci.	58
76653	Aoi	Elec. Eng.	60
98765	Bourikas	Elec. Eng.	98
98988	Tanaka	Biology	120

In [9]:
```
result = %sql select * from db_book.student
result.DataFrame()
```

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

Out[9]:

	ID	name	dept_name	tot_cred
0	00128	Zhang	Comp. Sci.	102
1	12345	Shankar	Comp. Sci.	32
2	19991	Brandt	History	80
3	23121	Chavez	Finance	110
4	44553	Peltier	Physics	56
5	45678	Levy	Physics	46
6	54321	Williams	Comp. Sci.	54
7	55739	Sanchez	Music	38
8	70557	Snow	Physics	0
9	76543	Brown	Comp. Sci.	58
10	76653	Aoi	Elec. Eng.	60
11	98765	Bourikas	Elec. Eng.	98
12	98988	Tanaka	Biology	120

In [ ]: