

# W4111\_HW0

September 14, 2021

## 1 COMS W4111: Introduction to Databases

### 1.1 Homework 0 - Environment Setup

#### 1.1.1 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 17, 10:00am ET 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!

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

```
name = "Aditya Kelvianto Sidharta"
uni = "aks2266"
track = "Programming"

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

```
Aditya Kelvianto Sidharta
aks2266
Programming
```

### 1.2 Anaconda

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

## Python Version

```
[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 < 5)):
    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=8, releaselevel='final', serial=0)

Your Python version is OK.

## Python Path

```
[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.")
```

/home/adityasidharta/columbia/database/hw0/hw0  
/home/adityasidharta/anaconda3/lib/python38.zip  
Found anaconda3  
Found some kind of Python.  
OK. Path is good.

Passed all path tests.

### Test Conda/Anaconda Version

```
[4]: import conda
```

```
[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=8, micro=8, releaselevel='final', serial=0)
Conda version information: sys.version_info(major=3, minor=8, micro=8,
releaselevel='final', serial=0)
```

Your Conda version is OK.

### Test Pandas

```
[6]: 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 1.2.4
Your version is OK.
```

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

```
[7]: !pip install pandas
```

```
Requirement already satisfied: pandas in
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (1.2.4)
Requirement already satisfied: python-dateutil>=2.7.3 in
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from pandas) (2.8.1)
Requirement already satisfied: pytz>=2017.3 in
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from pandas)
(2021.1)
Requirement already satisfied: numpy>=1.16.5 in
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from pandas)
```

(1.20.1)

Requirement already satisfied: six>=1.5 in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from python-  
dateutil>=2.7.3->pandas) (1.15.0)

## Install ipython-sql

```
[8]: !pip install ipython-sql
```

Requirement already satisfied: ipython-sql in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (0.4.0)  
Requirement already satisfied: six in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from ipython-sql)  
(1.15.0)  
Requirement already satisfied: ipython-genutils>=0.1.0 in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from ipython-sql)  
(0.2.0)  
Requirement already satisfied: sqlparse in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from ipython-sql)  
(0.4.2)  
Requirement already satisfied: sqlalchemy>=0.6.7 in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from ipython-sql)  
(1.4.15)  
Requirement already satisfied: ipython>=1.0 in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from ipython-sql)  
(7.22.0)  
Requirement already satisfied: prettytable<1 in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from ipython-sql)  
(0.7.2)  
Requirement already satisfied: pexpect>4.3 in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from  
ipython>=1.0->ipython-sql) (4.8.0)  
Requirement already satisfied: traitlets>=4.2 in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from  
ipython>=1.0->ipython-sql) (5.0.5)  
Requirement already satisfied: setuptools>=18.5 in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from  
ipython>=1.0->ipython-sql) (52.0.0.post20210125)  
Requirement already satisfied: pickleshare in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from  
ipython>=1.0->ipython-sql) (0.7.5)  
Requirement already satisfied: pygments in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from  
ipython>=1.0->ipython-sql) (2.8.1)  
Requirement already satisfied: prompt-toolkit!=3.0.0,!<3.0.1,<3.1.0,>=2.0.0 in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from  
ipython>=1.0->ipython-sql) (3.0.17)  
Requirement already satisfied: backcall in  
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from

```
ipython>=1.0->ipython-sql) (0.2.0)
Requirement already satisfied: decorator in
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from
ipython>=1.0->ipython-sql) (5.0.6)
Requirement already satisfied: jedi>=0.16 in
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from
ipython>=1.0->ipython-sql) (0.17.2)
Requirement already satisfied: parso<0.8.0,>=0.7.0 in
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from
jedi>=0.16->ipython>=1.0->ipython-sql) (0.7.0)
Requirement already satisfied: ptyprocess>=0.5 in
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from
pexpect>4.3->ipython>=1.0->ipython-sql) (0.7.0)
Requirement already satisfied: wcwidth in
/home/adityasidharta/anaconda3/lib/python3.8/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
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from
sqlalchemy>=0.6.7->ipython-sql) (1.0.0)
```

- 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.

```
[9]: %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

```
[10]: !pip install sqlalchemy
      !pip install pymysql
```

```
Requirement already satisfied: sqlalchemy in
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (1.4.15)
Requirement already satisfied: greenlet!=0.4.17 in
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (from sqlalchemy)
(1.0.0)
Requirement already satisfied: pymysql in
/home/adityasidharta/anaconda3/lib/python3.8/site-packages (1.0.2)
```

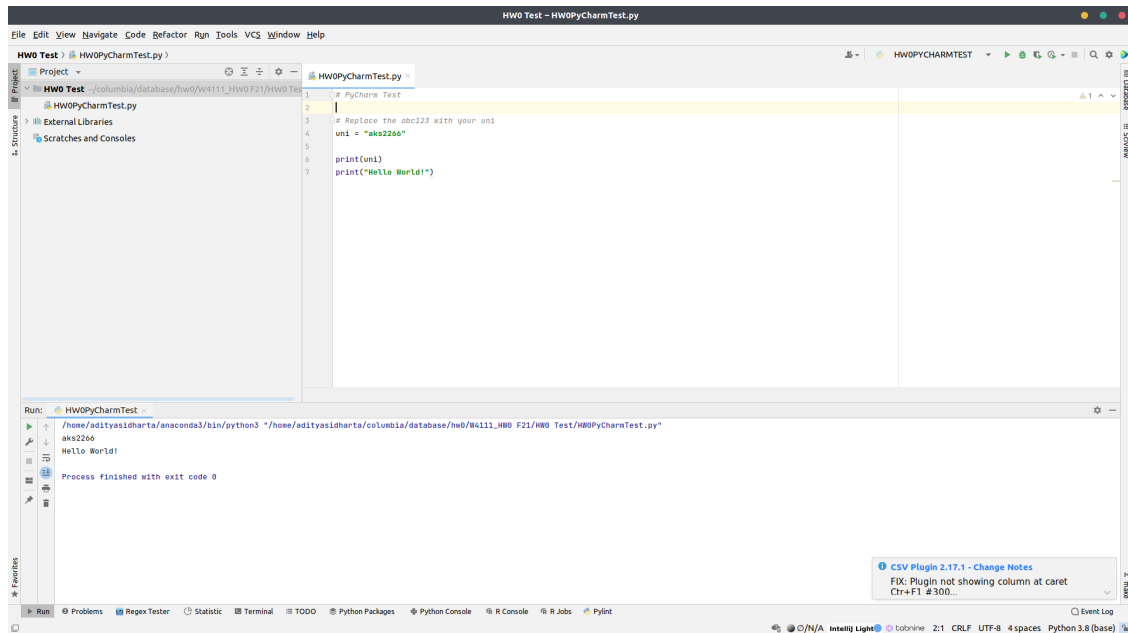
### 1.3 PyCharm

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.

```
[11]: from IPython.display import Image
```

```
Image("pycharm.png")
```

[11]:

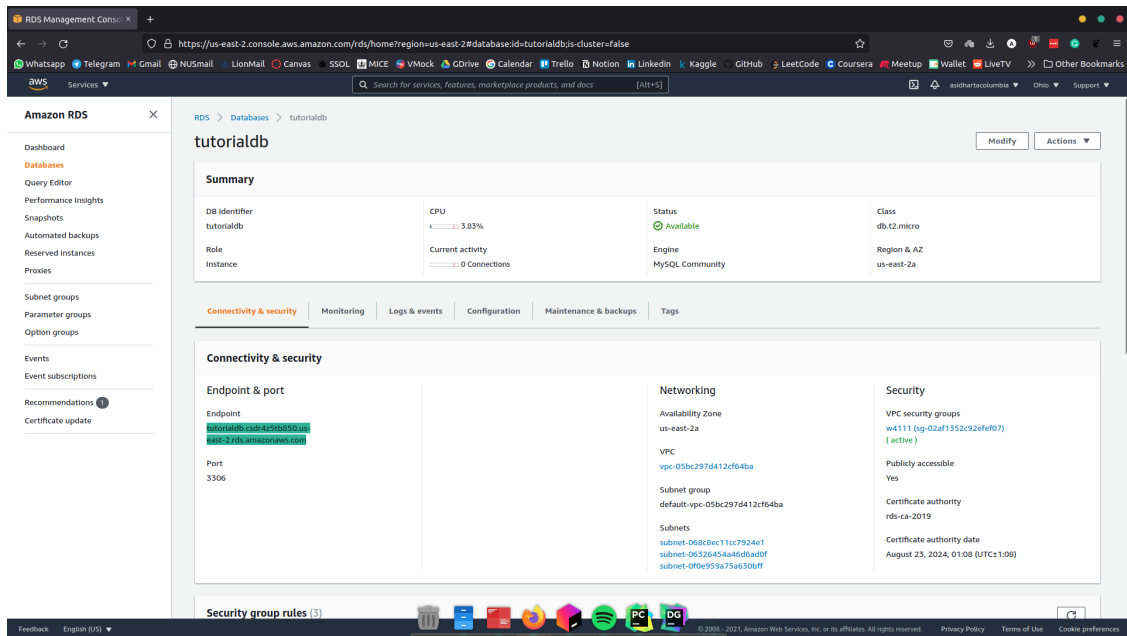


### 1.4 AWS

Follow the instructions to setup the AWS MySQL server. Insert your screenshot into the notebook using the cell below. You may have to change the path to the name and/or location of your image.

```
[12]: Image("aws.png")
```

[12]:

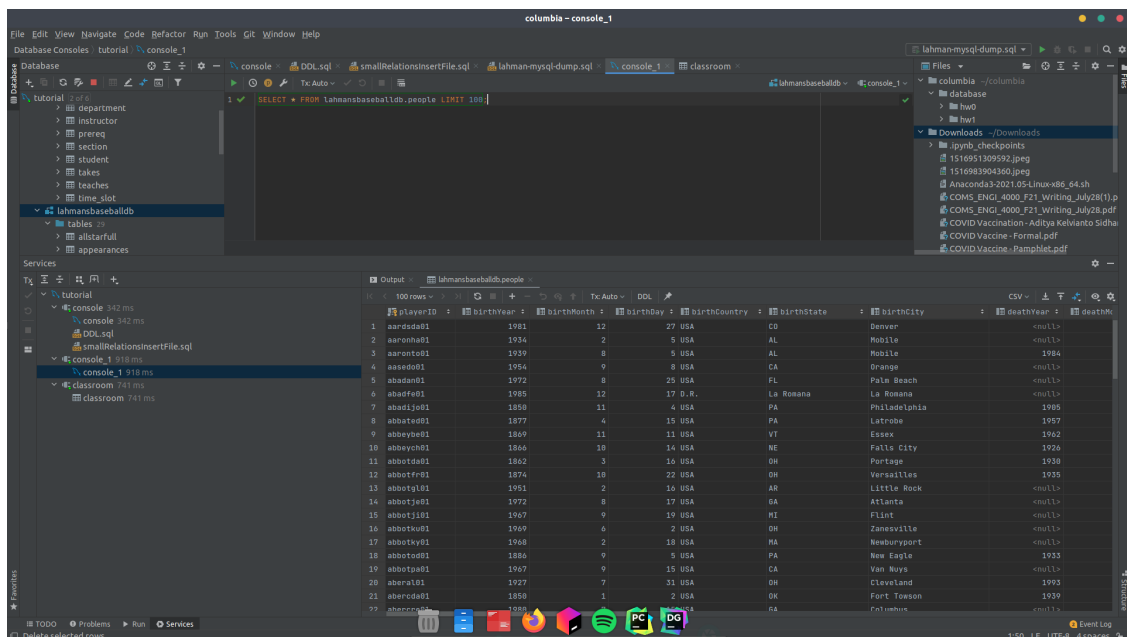


## 1.5 DataGrip

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.

[13]: `Image("datagrip.png")`

[13]:



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

```
[15]: %load_ext sql
      %sql mysql+pymysql://admin:password@tutorialdb.csdr4z5tb850.us-east-2.rds.
      ↪amazonaws.com/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:

```
[17]: %sql SELECT * FROM lahmansbaseballdb.people LIMIT 10;
```

```
* mysql+pymysql://admin:***@tutorialdb.csdr4z5tb850.us-
east-2.rds.amazonaws.com/lahmansbaseballdb
10 rows affected.
```

```
[17]: [('aardsda01', 1981, 12, 27, 'USA', 'CO', 'Denver', None, None, None, None,
None, None, 'David', 'Aardsma', 'David Allan', 215, 75, 'R', 'R', '2004-04-06',
'2015-08-23', 'aardd001', 'aardsda01', datetime.date(1981, 12, 27),
datetime.date(2004, 4, 6), datetime.date(2015, 8, 23), None),
('aaronha01', 1934, 2, 5, 'USA', 'AL', 'Mobile', None, None, None, None, None,
None, 'Hank', 'Aaron', 'Henry Louis', 180, 72, 'R', 'R', '1954-04-13',
'1976-10-03', 'aarah01', 'aaronha01', datetime.date(1934, 2, 5),
datetime.date(1954, 4, 13), datetime.date(1976, 10, 3), None),
('aaronto01', 1939, 8, 5, 'USA', 'AL', 'Mobile', 1984, 8, 16, 'USA', 'GA',
'Atlanta', 'Tommie', 'Aaron', 'Tommie Lee', 190, 75, 'R', 'R', '1962-04-10',
'1971-09-26', 'aarat01', 'aaronto01', datetime.date(1939, 8, 5),
datetime.date(1962, 4, 10), datetime.date(1971, 9, 26), datetime.date(1984, 8,
16)),
('aasedo01', 1954, 9, 8, 'USA', 'CA', 'Orange', None, None, None, None, None,
None, 'Don', 'Aase', 'Donald William', 190, 75, 'R', 'R', '1977-07-26',
'1990-10-03', 'aased001', 'aasedo01', datetime.date(1954, 9, 8),
datetime.date(1977, 7, 26), datetime.date(1990, 10, 3), None),
('abadan01', 1972, 8, 25, 'USA', 'FL', 'Palm Beach', None, None, None, None,
None, None, 'Andy', 'Abad', 'Fausto Andres', 184, 73, 'L', 'L', '2001-09-10',
'2006-04-13', 'abada001', 'abadan01', datetime.date(1972, 8, 25),
datetime.date(2001, 9, 10), datetime.date(2006, 4, 13), None),
('abadfe01', 1985, 12, 17, 'D.R.', 'La Romana', 'La Romana', None, None, None,
None, None, None, 'Fernando', 'Abad', 'Fernando Antonio', 220, 73, 'L', 'L',
'2010-07-28', '2019-09-28', 'abadf001', 'abadfe01', datetime.date(1985, 12, 17),
datetime.date(2010, 7, 28), datetime.date(2019, 9, 28), None),
('abadijo01', 1850, 11, 4, 'USA', 'PA', 'Philadelphia', 1905, 5, 17, 'USA',
'NJ', 'Pemberton', 'John', 'Abadie', 'John W.', 192, 72, 'R', 'R', '1875-04-26',
'1875-06-10', 'abadj01', 'abadijo01', datetime.date(1850, 11, 4),
datetime.date(1875, 4, 26), datetime.date(1875, 6, 10), datetime.date(1905, 5,
```



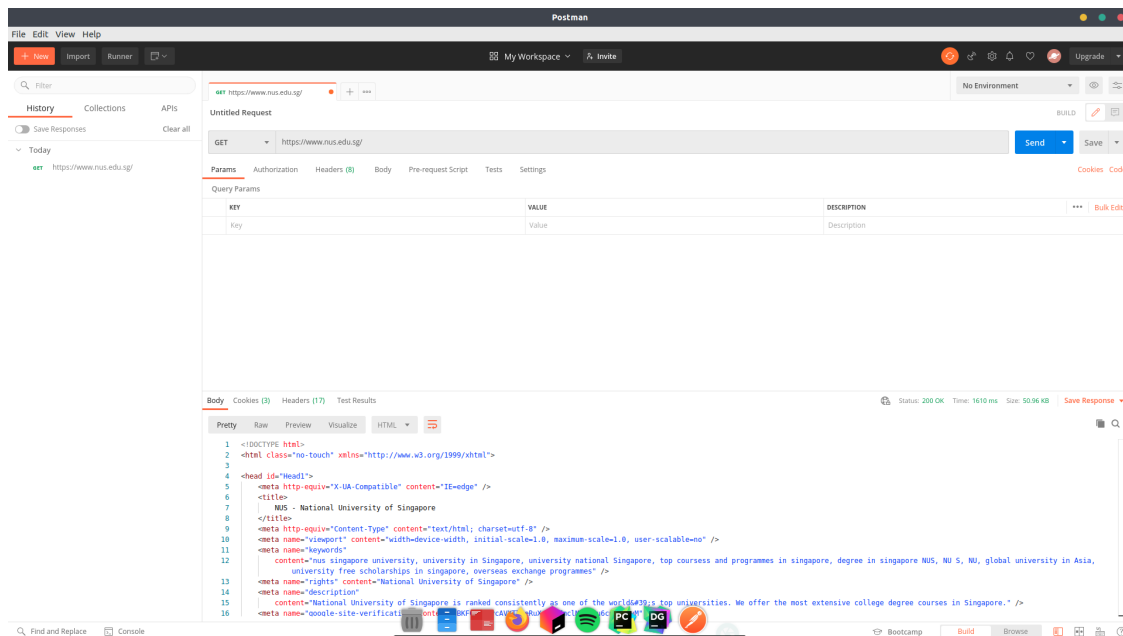
```
17)),
('abbated01', 1877, 4, 15, 'USA', 'PA', 'Latrobe', 1957, 1, 6, 'USA', 'FL',
'Fort Lauderdale', 'Ed', 'Abbatichio', 'Edward James', 170, 71, 'R', 'R',
'1897-09-04', '1910-09-15', 'abbae101', 'abbated01', datetime.date(1877, 4, 15),
datetime.date(1897, 9, 4), datetime.date(1910, 9, 15), datetime.date(1957, 1,
6)),
('abbeybe01', 1869, 11, 11, 'USA', 'VT', 'Essex', 1962, 6, 11, 'USA', 'VT',
'Colchester', 'Bert', 'Abbey', 'Bert Wood', 175, 71, 'R', 'R', '1892-06-14',
'1896-09-23', 'abbab101', 'abbeybe01', datetime.date(1869, 11, 11),
datetime.date(1892, 6, 14), datetime.date(1896, 9, 23), datetime.date(1962, 6,
11)),
('abbeych01', 1866, 10, 14, 'USA', 'NE', 'Falls City', 1926, 4, 27, 'USA',
'CA', 'San Francisco', 'Charlie', 'Abbey', 'Charles S.', 169, 68, 'L', 'L',
'1893-08-16', '1897-08-19', 'abbec101', 'abbeych01', datetime.date(1866, 10,
14), datetime.date(1893, 8, 16), datetime.date(1897, 8, 19), datetime.date(1926,
4, 27))]
```

## 1.6 Postman

Required for Programming Track only. Follow the instructions to setup Postman. Insert your screenshot of the successful GET request on the website you chose using the cell below. You may have to change the path to the name and/or location of your image.

[18]: `Image("postman.png")`

[18]:



[ ]: