COMS W4111: Introduction to Databases

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 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!

In [1]:

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

name = "Tushar"
uni = "tg2749"
track = "Programming"

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

```
Tushar
tg2749
Programming
```

Anaconda

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

Python 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 < 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=7, micro=3, re
leaselevel='final', serial=0)

Your Python version is OK.
```

Python Path

In [22]:

```
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:\PersonalFiles\Courses\IntroToDB\HW0\W4111_HW0_F21\W4111_HW0 F21
C:\Users\tushar\Anaconda3\python37.zip
Found anaconda3
Found some kind of Python.
OK. Path is good.

Passed all path tests.
```

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=7, micro=3, releaselevel='final', serial=
0)
Conda version information: sys.version_info(major=3, minor=7, micro=3, rel
easelevel='final', serial=0)

Your Conda version is OK.
```

Test Pandas

In [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
```

```
C:\Users\tushar\Anaconda3\lib\site-packages\pandas\compat\_optional.py:13
8: UserWarning: Pandas requires version '2.7.0' or newer of 'numexpr' (ver
sion '2.6.9' currently installed).
  warnings.warn(msg, UserWarning)

Your pandas version is  1.3.0
Your version is OK.
```

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

In [23]:

```
!pip install pandas
```

```
Requirement already satisfied: pandas in c:\users\tushar\anaconda3\lib\sit
e-packages (1.3.0)
Requirement already satisfied: numpy>=1.17.3 in c:\users\tushar\anaconda3
\lib\site-packages (from pandas) (1.21.1)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\tushar\a
naconda3\lib\site-packages (from pandas) (2.8.0)
Requirement already satisfied: pytz>=2017.3 in c:\users\tushar\anaconda3\l
ib\site-packages (from pandas) (2018.9)
Requirement already satisfied: six>=1.5 in c:\users\tushar\appdata\roaming
\python\python37\site-packages (from python-dateutil>=2.7.3->pandas) (1.1
5.0)
```

Install ipython-sql

In [24]:

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

```
Requirement already satisfied: ipython-sql in c:\users\tushar\anaconda3\li
b\site-packages (0.4.0)
Requirement already satisfied: prettytable<1 in c:\users\tushar\anaconda3
\lib\site-packages (from ipython-sql) (0.7.2)
Requirement already satisfied: sqlparse in c:\users\tushar\anaconda3\lib\s
ite-packages (from ipython-sql) (0.4.2)
Requirement already satisfied: sqlalchemy>=0.6.7 in c:\users\tushar\anacon
da3\lib\site-packages (from ipython-sql) (1.3.1)
Requirement already satisfied: ipython>=1.0 in c:\users\tushar\anaconda3\l
ib\site-packages (from ipython-sql) (7.4.0)
Requirement already satisfied: ipython-genutils>=0.1.0 in c:\users\tushar
\anaconda3\lib\site-packages (from ipython-sql) (0.2.0)
Requirement already satisfied: six in c:\users\tushar\appdata\roaming\pyth
on\python37\site-packages (from ipython-sql) (1.15.0)
Requirement already satisfied: backcall in c:\users\tushar\anaconda3\lib\s
ite-packages (from ipython>=1.0->ipython-sql) (0.1.0)
Requirement already satisfied: colorama; sys_platform == "win32" in c:\use
rs\tushar\anaconda3\lib\site-packages (from ipython>=1.0->ipython-sql) (0.
4.1)
Requirement already satisfied: decorator in c:\users\tushar\anaconda3\lib
\site-packages (from ipython>=1.0->ipython-sql) (4.4.0)
Requirement already satisfied: prompt-toolkit<2.1.0,>=2.0.0 in c:\users\tu
shar\anaconda3\lib\site-packages (from ipython>=1.0->ipython-sql) (2.0.9)
Requirement already satisfied: pygments in c:\users\tushar\anaconda3\lib\s
ite-packages (from ipython>=1.0->ipython-sql) (2.3.1)
Requirement already satisfied: setuptools>=18.5 in c:\users\tushar\appdata
\roaming\python\python37\site-packages (from ipython>=1.0->ipython-sql) (5
6.0.0)
Requirement already satisfied: traitlets>=4.2 in c:\users\tushar\anaconda3
\lib\site-packages (from ipython>=1.0->ipython-sql) (4.3.2)
Requirement already satisfied: jedi>=0.10 in c:\users\tushar\anaconda3\lib
\site-packages (from ipython>=1.0->ipython-sql) (0.13.3)
Requirement already satisfied: pickleshare in c:\users\tushar\anaconda3\li
b\site-packages (from ipython>=1.0->ipython-sql) (0.7.5)
Requirement already satisfied: wcwidth in c:\users\tushar\anaconda3\lib\si
te-packages (from prompt-toolkit<2.1.0,>=2.0.0->ipython>=1.0->ipython-sql)
(0.1.7)
Requirement already satisfied: parso>=0.3.0 in c:\users\tushar\anaconda3\l
ib\site-packages (from jedi>=0.10->ipython>=1.0->ipython-sql) (0.3.4)
```

- If you got errors, please follow the instructions in the ipython-sql site (https://github.com/catherinedevlin/ipython-sql) 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 [8]:

```
%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

In [31]:

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

```
Requirement already satisfied: sqlalchemy in c:\users\tushar\anaconda3\lib
\site-packages (1.3.1)
Requirement already satisfied: pymysql in c:\users\tushar\anaconda3\lib\si
te-packages (1.0.2)
```

PyCharm

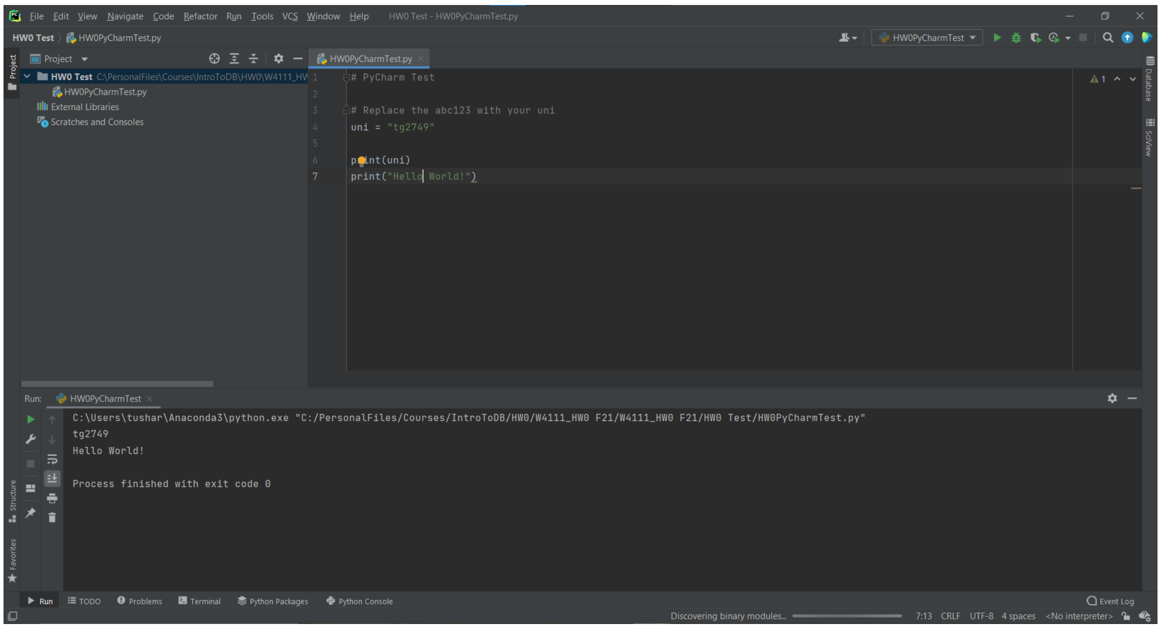
Required for Programming Track only, but recommended for all. Follow the instructions to setup PyCharm and run the test. Take a screenshot and insert it into the notebook using the cell below. You may have to change the path to the name and/or location of your image.

In [10]:

```
from IPython.display import Image

Image("PycharmScreenshot.png")
```

Out[10]:


File Edit View Navigate Code Refactor Run Tools VCS Window Help HW0 Test - HW0PyCharmTest.py
HW0 Test HW0PyCharmTest.py
HW0PyCharmTest
Project
HW0 Test C:\PersonalFiles\Courses\IntroToDB\HW0\W4111_HW
HW0PyCharmTest.py
External Libraries
Scratches and Consoles
Run: HW0PyCharmTest
Run TODO Problems Terminal Python Packages Python Console Event Log
Discovering binary modules... 7:13 CRLF UTF-8 4 spaces <No interpreter>

```
# PyCharm Test

# Replace the abc123 with your uni
uni = "tg2749"

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

```
C:\Users\tushar\Anaconda3\python.exe "C:/PersonalFiles/Courses/IntroToDB/HW0/W4111_HW0 F21/W4111_HW0 F21/HW0 Test/HW0PyCharmTest.py"
tg2749
Hello World!

Process finished with exit code 0
```

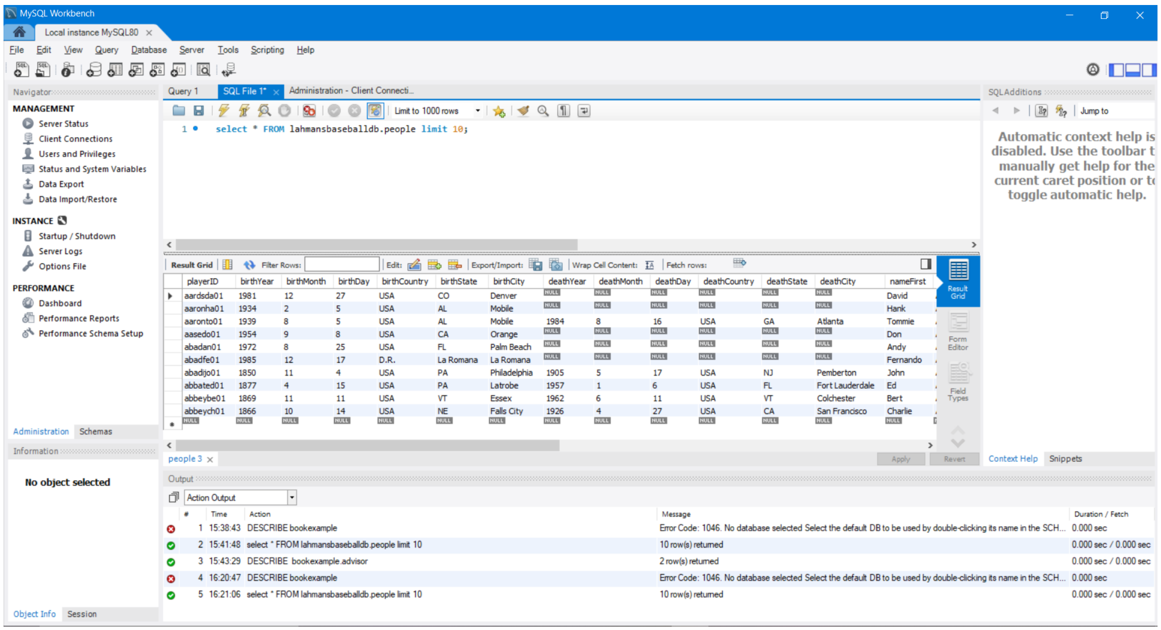
MySQL server

Follow the instructions to setup the 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.

In [38]:

```
Image("MySQLScreenshot.png")
```

Out[38]:



```
select * FROM lahmansbaseballdb.people limit 10;
```

playerID	birthYear	birthMonth	birthDay	birthCountry	birthState	birthCity	deathYear	deathMonth	deathDay	deathCountry	deathState	deathCity	nameFirst
aardsda01	1981	12	27	USA	CO	Denver	NULL	NULL	NULL	NULL	NULL	NULL	David
aaronha01	1934	2	5	USA	AL	Mobile	NULL	NULL	NULL	NULL	NULL	NULL	Hank
aaronto01	1939	8	5	USA	AL	Mobile	1984	8	16	USA	GA	Atlanta	Tommie
aasedo01	1954	9	8	USA	CA	Orange	NULL	NULL	NULL	NULL	NULL	NULL	Don
abadan01	1972	8	25	USA	FL	Palm Beach	NULL	NULL	NULL	NULL	NULL	NULL	Andy
abadfe01	1985	12	17	D.R.	La Romana	La Romana	NULL	NULL	NULL	NULL	NULL	NULL	Fernando
abadijo01	1850	11	4	USA	PA	Philadelphia	1905	5	17	USA	NJ	Pemberton	John
abbated01	1877	4	15	USA	PA	Latrobe	1957	1	6	USA	FL	Fort Lauderdale	Ed
abbeybe01	1869	11	11	USA	VT	Essex	1962	6	11	USA	VT	Colchester	Bert
abbeych01	1866	10	14	USA	NE	Falls City	1926	4	27	USA	CA	San Francisco	Charlie
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
#  Time      Action                                         Message                                                                                    Duration / Fetch
1  15:38:43  DESCRIBE bookexample                           Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...  0.000 sec
2  15:41:48  select * FROM lahmansbaseballdb.people limit 10  10 row(s) returned                                                                    0.000 sec / 0.000 sec
3  15:43:29  DESCRIBE  bookexample.advisor                  2 row(s) returned                                                                           0.000 sec / 0.000 sec
4  16:20:47  DESCRIBE bookexample                           Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the SCH...  0.000 sec
5  16:21:06  select * FROM lahmansbaseballdb.people limit 10  10 row(s) returned                                                                    0.000 sec / 0.000 sec
```

Automatic context help is disabled. Use the toolbar t manually get help for the current caret position or t toggle automatic help.

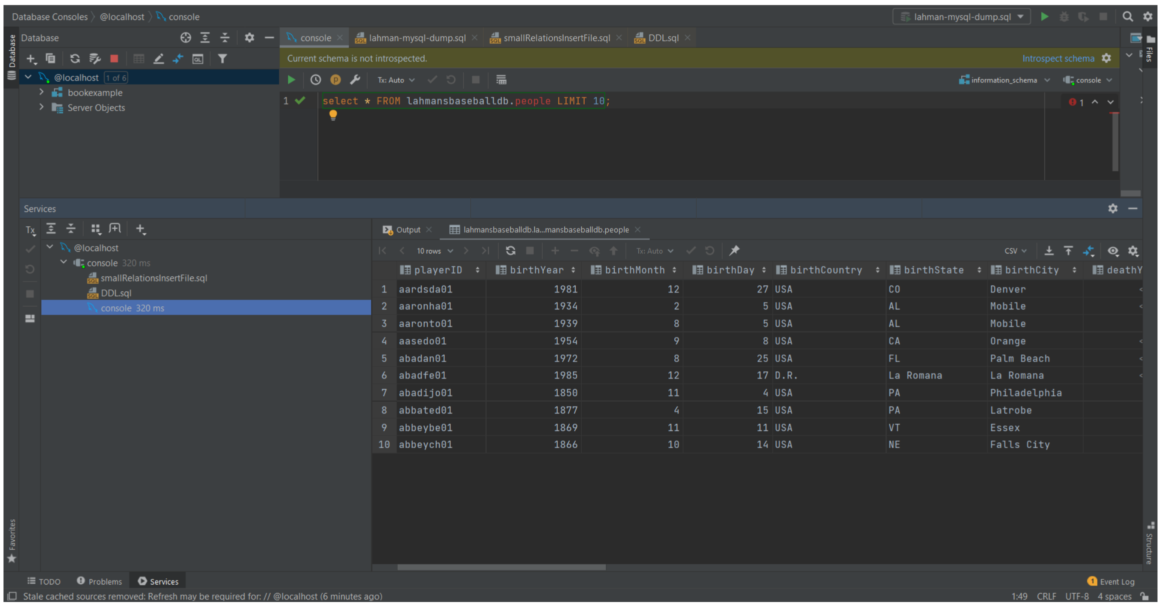
DataGrip

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 [11]:

```
Image("DatagripScreenshot.png")
```

Out[11]:



```
select * FROM lahmansbaseballdb.people LIMIT 10;
```

	playerID	birthYear	birthMonth	birthDay	birthCountry	birthState	birthCity
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 Beach
6	abadfe01	1985	12	17	D.R.	La Romana	La Romana
7	abadijo01	1850	11	4	USA	PA	Philadelphia
8	abbated01	1877	4	15	USA	PA	Latrobe
9	abbeybe01	1869	11	11	USA	VT	Essex
10	abbeych01	1866	10	14	USA	NE	Falls City

Stale cached sources removed: Refresh may be required for: // @localhost (6 minutes ago)

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 [25]:

```
%load_ext sql
%reload_ext sql
```

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

In [34]:

```
%sql mysql+pymysql://root:admin123@localhost:3306/lahmansbaseballdb
```

In [21]:

```
import pymysql

import pandas as pd

conn=pymysql.connect(host='localhost',port=int(3306),user='root',passwd='admin123')

df=pd.read_sql_query("select * FROM lahmansbaseballdb.people LIMIT 10;",conn)

print(df)
```

```
    playerID  birthYear  birthMonth  birthDay birthCountry birthState  \
0  aardsda01       1981          12        27          USA         CO
1  aaronha01       1934           2         5          USA         AL
2  aaronto01       1939           8         5          USA         AL
3   aasedo01       1954           9         8          USA         CA
4   abadan01       1972           8        25          USA         FL
5   abadfe01       1985          12        17         D.R.  La Romana
6  abadijo01       1850          11         4          USA         PA
7  abbated01       1877           4        15          USA         PA
8  abbeybe01       1869          11        11          USA         VT
9  abbeych01       1866          10        14          USA         NE

      birthCity  deathYear  deathMonth  deathDay  ... bats throws       de
but  \
0        Denver        NaN         NaN       NaN  ...    R      R  2004-04
-06
1        Mobile        NaN         NaN       NaN  ...    R      R  1954-04
-13
2        Mobile     1984.0         8.0      16.0  ...    R      R  1962-04
-10
3        Orange        NaN         NaN       NaN  ...    R      R  1977-07
-26
4    Palm Beach        NaN         NaN       NaN  ...    L      L  2001-09
-10
5     La Romana        NaN         NaN       NaN  ...    L      L  2010-07
-28
6  Philadelphia     1905.0         5.0      17.0  ...    R      R  1875-04
-26
7       Latrobe     1957.0         1.0       6.0  ...    R      R  1897-09
-04
8         Essex     1962.0         6.0      11.0  ...    R      R  1892-06
-14
9    Falls City     1926.0         4.0      27.0  ...    L      L  1893-08
-16

    finalGame   retroID    bbrefID  birth_date  debut_date finalgame_date
\
0  2015-08-23  aardd001  aardsda01  1981-12-27  2004-04-06     2015-08-23
1  1976-10-03  aaroh101  aaronha01  1934-02-05  1954-04-13     1976-10-03
2  1971-09-26  aarot101  aaronto01  1939-08-05  1962-04-10     1971-09-26
3  1990-10-03  aased001   aasedo01  1954-09-08  1977-07-26     1990-10-03
4  2006-04-13  abada001   abadan01  1972-08-25  2001-09-10     2006-04-13
5  2019-09-28  abadf001   abadfe01  1985-12-17  2010-07-28     2019-09-28
6  1875-06-10  abadj101  abadijo01  1850-11-04  1875-04-26     1875-06-10
7  1910-09-15  abbae101  abbated01  1877-04-15  1897-09-04     1910-09-15
8  1896-09-23  abbeb101  abbeybe01  1869-11-11  1892-06-14     1896-09-23
9  1897-08-19  abbec101  abbeych01  1866-10-14  1893-08-16     1897-08-19

   death_date
0        None
1        None
2  1984-08-16
3        None
4        None
5        None
6  1905-05-17
7  1957-01-06
8  1962-06-11
9  1926-04-27

[10 rows x 28 columns]
```

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

The below command ran into issues hence used pymysql connect to return the query results

In [37]:

```
%sql select * from lahmansbaseballdb.appearances limit 10;
```

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

```
---------------------------------------------------------------------------
-
KeyError                                  Traceback (most recent call las
t)
<ipython-input-37-2dd6cc0dd177> in <module>
----> 1 get_ipython().run_line_magic('sql', 'select * from lahmansbaseball
db.appearances limit 10;')

~\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py in run_line
_magic(self, magic_name, line, _stack_depth)
   2305                 kwargs['local_ns'] = sys._getframe(stack_depth).f_
locals
   2306             with self.builtin_trap:
-> 2307                 result = fn(*args, **kwargs)
   2308             return result
   2309 

<C:\Users\tushar\Anaconda3\lib\site-packages\decorator.py:decorator-gen-12
8> in execute(self, line, cell, local_ns)

~\Anaconda3\lib\site-packages\IPython\core\magic.py in <lambda>(f, *a, **
k)
    185     # but it's overkill for just that one bit of state.
    186     def magic_deco(arg):
--> 187         call = lambda f, *a, **k: f(*a, **k)
    188 
    189         if callable(arg):

<C:\Users\tushar\Anaconda3\lib\site-packages\decorator.py:decorator-gen-12
7> in execute(self, line, cell, local_ns)

~\Anaconda3\lib\site-packages\IPython\core\magic.py in <lambda>(f, *a, **
k)
    185     # but it's overkill for just that one bit of state.
    186     def magic_deco(arg):
--> 187         call = lambda f, *a, **k: f(*a, **k)
    188 
    189         if callable(arg):

~\Anaconda3\lib\site-packages\sql\magic.py in execute(self, line, cell, lo
cal_ns)
    215 
    216         try:
--> 217             result = sql.run.run(conn, parsed["sql"], self, user_n
s)
    218 
    219             if (

~\Anaconda3\lib\site-packages\sql\run.py in run(conn, sql, config, user_na
mespace)
    369             if result and config.feedback:
    370                 print(interpret_rowcount(result.rowcount))
--> 371         resultset = ResultSet(result, statement, config)
    372         if config.autopandas:
    373             return resultset.DataFrame()

~\Anaconda3\lib\site-packages\sql\run.py in __init__(self, sqlaproxy, sql,
config)
    110         self.limit = config.autolimit
    111         style_name = config.style
--> 112         self.style = prettytable.__dict__[style_name.upper()]
```

```
    113         if sqlaproxy.returns_rows:
    114             if self.limit:

KeyError: 'DEFAULT'
```

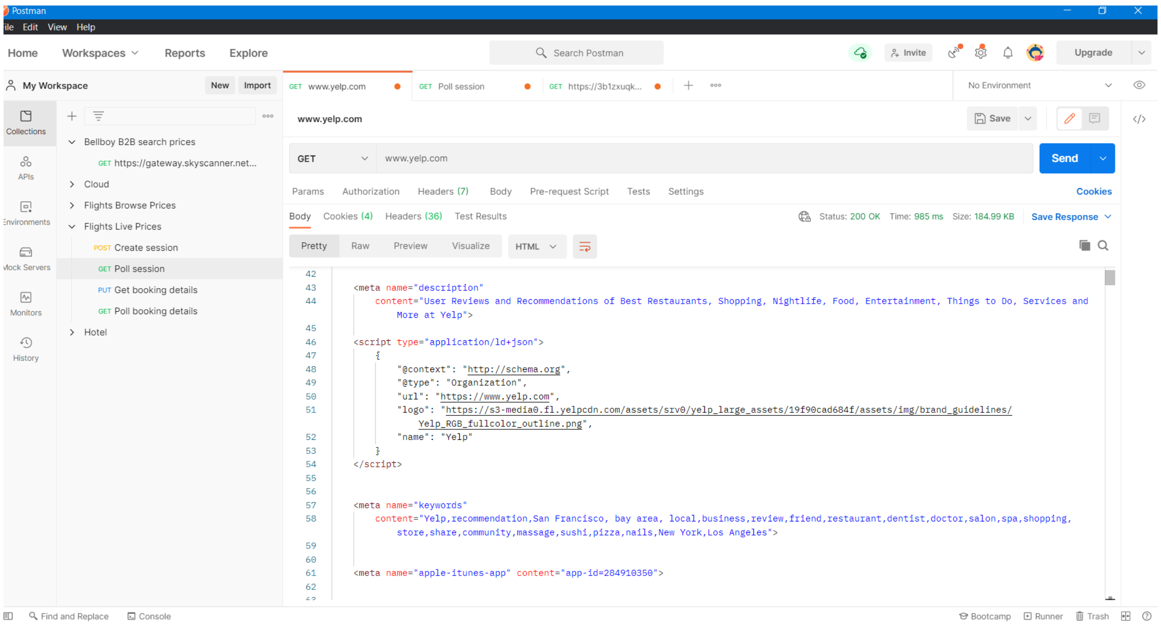
Postman

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.

In [12]:

```
Image("PostmanScreenshot.png")
```

Out[12]:



Status: 200 OK Time: 985 ms Size: 184.99 KB

```

<meta name="description"
    content="User Reviews and Recommendations of Best Restaurants, Shopping, Nightlife, Food, Entertainment, Things to Do, Services and
        More at Yelp">

<script type="application/ld+json">
    {
        "@context": "http://schema.org",
        "@type": "Organization",
        "url": "https://www.yelp.com",
        "logo": "https://s3-media0.fl.yelpcdn.com/assets/srv0/yelp_large_assets/19f90cad684f/assets/img/brand_guidelines/
            Yelp_RGB_fullcolor_outline.png",
        "name": "Yelp"
    }
</script>


<meta name="keywords"
    content="Yelp,recommendation,San Francisco, bay area, local,business,review,friend,restaurant,dentist,doctor,salon,spa,shopping,
        store,share,community,massage,sushi,pizza,nails,New York,Los Angeles">


<meta name="apple-itunes-app" content="app-id=284910350">

```