Integrating MySQL Table Data with a Python Script

John Wensink

ITS410 Database Management

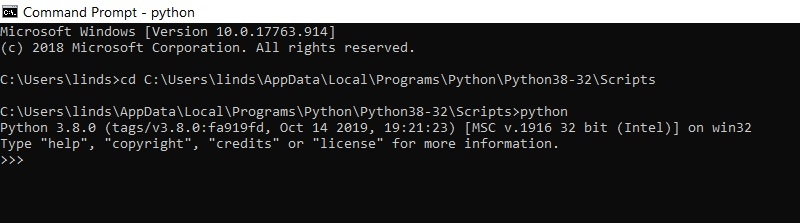
Colorado State University-Global Campus

Dr. Dan Morrill

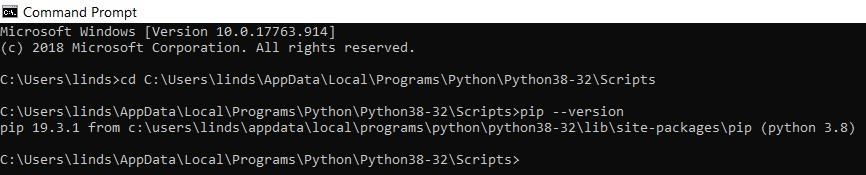
August 6, 2018

Integrating MySQL Table Data with a Python Script

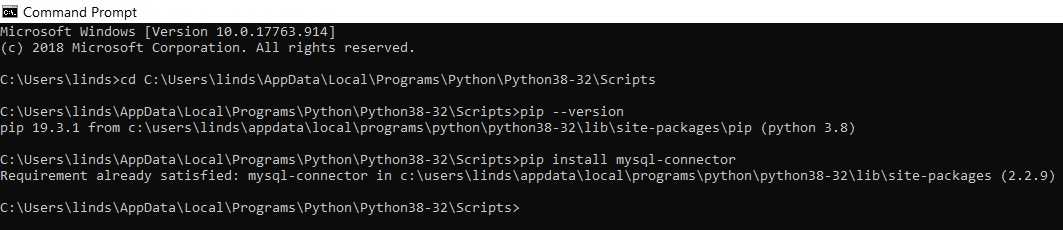
For this project, we will use the Connector/Python library from Oracle. A benefit of using this library is its complete implementation of communication protocols used by the MySQL RDBMS. This library is PEP 249 compliant and can be used alongside other APIs ensuring standardized interfacing between commonly available python modules used to access relational databases (Python, n.d.). In Python, packages are installed and managed through a package management system called ‘Python Package Index’ (PIP). In order to install PIP, first, we must create a path to Python in Window’s environment settings by adding Python’s file location as a new path:



Next, we will open Window’s command prompt (as an administrator) and use the ‘**cd**’ command to use the directory where Python’s default scripts are stored. Now that we are using the correct directory the command ‘**pip install xlwt**’ (Patel, 2017) will install the packages required to use PIP within Python:



With PIP installed we are able to move on to installing MySQL/Connector. It is important to make sure your versions are [compatible](https://dev.mysql.com/doc/connector-python/en/connector-python-versions.html). In Windows command prompt execute the command ‘**pip install mysql-connector**’. The package will install:

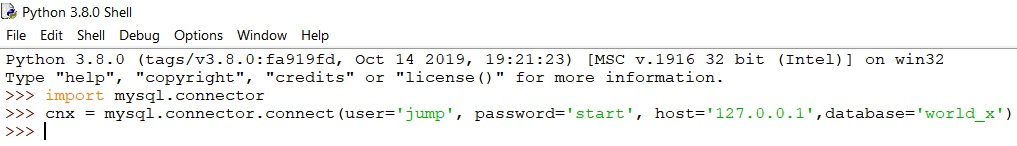


Now we will connect to the ‘jump’ server to our python shell. Using the Python shell, we will use the command ‘**import mysql.connector**’(Sen, 2017) to import the module into our Python terminal. We will title our connection ‘**cnx**’.Next, we will want to connect to the ‘world-x’ database stored in the ‘jump’ server by using the command:

**cnx = mysql.connector.connect**

**(user='jump',password='start',host='127.0.0.1',database='world\_x')** (MySQL, 2019).

If you did not return an error message you are now connected with the World-X database thru Python and can start performing queries and operations on the tables stored within:



**What if Population Were 10% Greater?**

Now that our programs are connected, we are ready to begin meaningful operations on our database tables by using Python commands. We will need to login to MySQL as ‘root’ now in order to grant privileges the user ‘jump’ on ‘localhost’ using the command:

**GRANT ALL ON world\_x.\* TO ‘jump’@’localhost’**

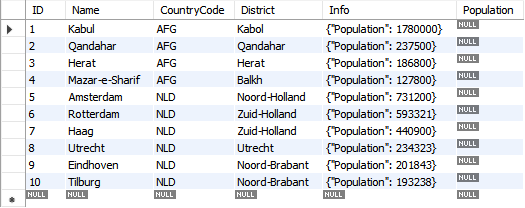
This will allow the user ‘jump’ to modify the tables for this assignment. In MySQL, the following command is entered to add a column called ‘Population’ to our ‘city’ table in the ‘world-x’ database:

**ALTER TABLE `world\_x`.`city`**

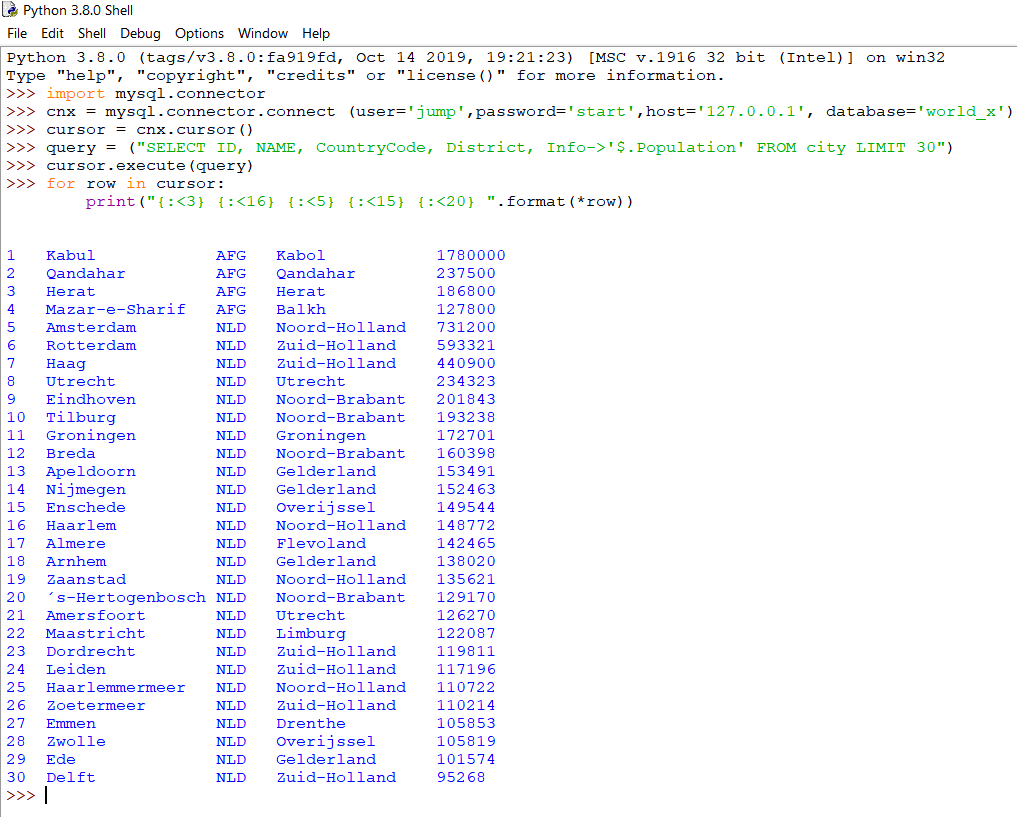
**ADD COLUMN `Population` DOUBLE NULL DEFAULT NULL AFTER `Info`;**

(CSU Global, n.d.)

We can see that a new column has been added to the ‘city’ table for us to store our newly created data.



Remember, the ‘Info’ column has been populated with ‘JavaScript Object Notation’ (JSON) data which is useful for storing name/value pairs (SecretGeek, 2006). The next step is to obtain a cursor in Python for which we will use the command ‘**cursor = cnx.cursor()’** (Boronczyk, 2015). From here we can use a simple select statement to make sure the connector will be returning the desired results. Formatting can be adjusted based on the character length in its field.



Everything seems to be performing nominally so we will continue to INSERT INTO the population using an SQL query in python that copies population data and increases it by 10% before inserting it into the population column. It never hurts to create a SAVEPOINT, we will call it ‘population’. We will need to disable safe updates mode by entering:

**query = (“SET SQL\_SAFE\_UPDATES = 0”)**

(Stormwild, 2013).

Now onto the INSERT statement. Let’s do this in two steps. First, we will copy the populations from the ‘Info’ column and insert them into the ‘Population’ column using the command:

**query = ("UPDATE city SET Population = info->'$.Population'")**

**cursor.execute(query)**

This returns without an error and we are able to move on to the second part of this operation, increasing the value of ‘Population’ by 10%. This is accomplished using the command:

**query = ("UPDATE city SET Population=(Population+(Population\*1/10))")**

**cursor.execute(query)**

Now we will COMMIT the UPDATE

**query = ("COMMIT")**

**cursor.execute(query)**

Let’s not forget to turn safe update mode back on:

**query = (“SET SQL\_SAFE\_UPDATES = 1”)**

(Stormwild, 2013)

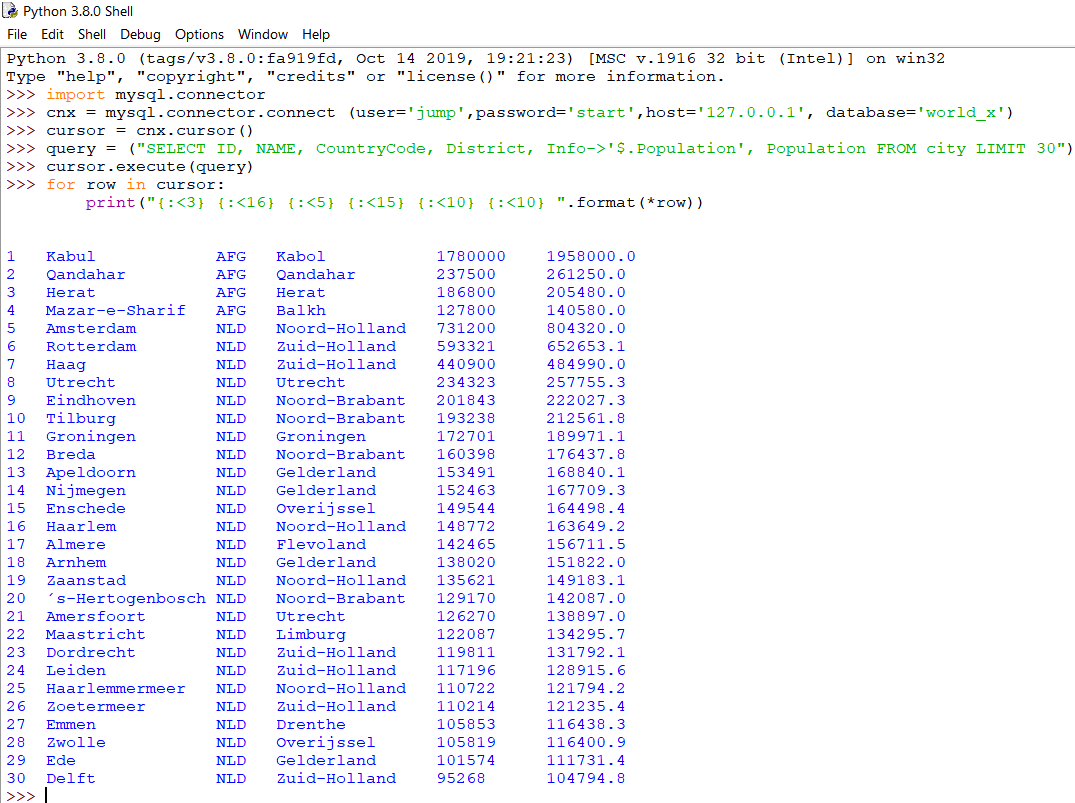
We should be good to view the changes now. Let’s run a select statement:

**query = ("SELECT ID, Name, CountryCode, District, Info->'$.Population', Population FROM city LIMIT 30")**

**cursor.execute(query)**

**for row in cursor:**

**print("{:<3} {:<16} {:<5} {:<15} {:<10} {:<10} ".format(\*row))**

****

References

Boronczyk, T. (2015). Jump Start MySQL. Retrieved from <https://bookshelf.vitalsource.com/#/books/9781457192838/cfi/96!/4/4@0.00:0.00>

CSU Global. (n.d.). Module 6: Critical Thinking. Retrieved December 22, 2019, from <https://csuglobal.instructure.com/courses/15481/assignments/309531?module_item_id=751819>.

MySQL. (2019). MySQL Connector/Python Developer Guide: 5.1 Connecting to MySQL Using Connector/Python. Retrieved December 22, 2019, from <https://dev.mysql.com/doc/connector-python/en/connector-python-example-connecting.html>.

*PIP Install Command in Python 3.6*. (2017). Retrieved from <https://www.youtube.com/watch?v=237dNNQhD3Q>

Python Developer's Guide: PEP Index. (n.d.). PEP 249 -- Python Database API Specification v2.0. Retrieved December 22, 2019, from <https://www.python.org/dev/peps/pep-0249/>.

SecretGeek. (2006, July 22). What is JSON: The 3 Minute JSON Tutorial. Retrieved from <http://www.secretgeek.net/json_3mins>.

Sen, P. (2017, June 8). How to Install MySQL-Connector via PIP. Retrieved December 22, 2019, from <https://stackoverflow.com/questions/32754461/how-to-install-mysql-connector-via-pip>.

Stormwild. (2013, June 10). How to Disable MySQL Safe Mode. Retrieved December 22, 2019, from <https://gist.github.com/stormwild/5746860>.