Python DB-API

Data Manipulation in Python

Georgia Tech

Database Programming in Python

- DB-API: https://www.python.org/dev/peps/pep-0249/
- SQLite3 is built-in: `import sqlite3`
- MySQL reguires third-party library

```
$ conda install pymysql
...
$ python
Python 3.6.0 |Continuum Analytics, Inc.| (default, Dec 23 2016, 13:19:00)
[GCC 4.2.1 Compatible Apple LLVM 6.0 (clang-600.0.57)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> import pymysql
```

Key point: most database APIs, including Python's DB-API, are simply ways of executing SQL statements and getting the results of SQL statements. You can't use DB-API without knowing SQL.

Georgia Tech

Working With Databases

- A connection objects represent a connection to a database

```
connection = pymysql.connect(...)
```

- Cursor object is a stateful pointer to a part of the database

```
cursor = connection.cursor()
```

- SQL Statements are submitted to cursor's `execute()` method
  – `execute` returns the number of rows in the statement's result

```
>>> cursor.execute('insert into table values (%s, %s)', ('field1', 'field2'))
1
```

  – If the statement was a select, then the cursor points at the first row of the result
    - The cursor object is an iterator over the results (preferred method)
    - `fetchall()` , `fetchone()` , etc. return results in Python data structures

```
>>> cursor.execute('select * from table where column1 = %s', ('field1'))
1
>>> for row in cursor: print(row)
{'column1': 'field1', 'column2': 'field'2'}
```

Georgia Tech

Connecting to a MySQL Database

If you configured your MySQL server without a root password, this will work:

```
>>> import pymysql
>>> connection = pymysql.connect(host='localhost',
                                 user='root',
                                 password='',
                                 db='pubs',
                                 charset='utf8mb4',
                                 cursorclass=pymysql.cursors.DictCursor)
```

- Substitute your own root password if you have one
- Notice that the cursor class id `DictCursor` - pymysql will return rows of dabases as dictionaries

Georgia Tech

Inserting Data into a Database Table

```
>>> cursor = connection.cursor()
>>> cursor.execute('insert into author values (%s, %s, %s)', (10, 'Jenny',
     'McCarthy'))
1
```

Executing Queries on a MySQL Database

```
>>> cursor = connection.cursor()
>>> query = "select * from author where last_name = %s"
>>> cursor.execute(query, ('McCarthy'))
2
>>> for row in cursor: print(row)
{'author_id': 1, 'first_name': 'John', 'last_name': 'McCarthy'}
{'author_id': 10, 'first_name': 'Jenny', 'last_name': 'McCarthy'}
```

- In the query string, use placeholders with same syntax as % - based string interpolation
- In call to `cursor.execute`, supply values for placeholders in a tuple.
- After calling `cursor.execute`, cursor is an iterator over the result rows.

Georgia Tech