CS 2316 Data Manipulation for Engineers

PyMySQL

Christopher Simpkins
chris.simpkins@gatech.edu

MySQL

MySQL is the most popular open source database.

- Client-server
- Advanced features
- Used by many top web sites, like Facebook, Wikipedia, many more

In this class we will use a remote MySQL server via Python's pymysql module. First I'll show you how to install it and use it on your own system if you're so inclined (optional).

The CS 2316 MySQL Server

First you need to install MySQL.

- On Mac OSX, Homebrew is probably the easiest way
- On linux use your pacakage manager
- On Windows download an installer from http://www.mysql.com/

The MySQL shell is similar to SQLite's. Supply a hostname, user name, and password (given to you via email).

> *DO NOT GIVE OUT THE PASSWORD! THIS DATABASE IS FOR CS2316 ONLY!*

NOTE: You must be on the Georgia Tech network (on campus or VPN) to access the class MySQL server.

```
$ mysql -h academic-mysql.cc.gatech.edu -u cs2316 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11844
Server version: 5.1.73 Source distribution
...
mysql>
```

The MySQL Shell

Instead of "dot" commands, MySQL uses slashes:

```
mysql> \help
...
List of all MySQL commands:
Note that all text commands must be first on line and end with ';'
?         (\?) Synonym for `help'.
clear     (\c) Clear the current input statement.
connect   (\r) Reconnect to the server. Optional arguments are db and
    host.
...
```

To use a database, you'll need to issue the `use` command:

```
mysql> use cs2316db;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

The CS 2316 Database

You can see all the tables with `show tables`

```
mysql> show tables;
+----------------------+
| Tables_in_cs2316db   |
+----------------------+
| BrokerageTransaction |
| BrokerageUsers       |
...
```

Get a table schema with `describe`

```
mysql> describe PizzaOrders;
+-----------+----------+------+-----+---------+----------------+
| Field     | Type     | Null | Key | Default | Extra          |
+-----------+----------+------+-----+---------+----------------+
| OrderID   | int(11)  | NO   | PRI | NULL    | auto_increment |
| User      | text     | NO   |     | NULL    |                |
| Cheese    | int(11)  | NO   |     | NULL    |                |
| Pepperoni | int(11)  | NO   |     | NULL    |                |
| Veggie    | int(11)  | NO   |     | NULL    |                |
+-----------+----------+------+-----+---------+----------------+
```

Identifiers are case-sensitive. `pizzaorders != PizzaOrders`

PyMySQL

Install PyMySQL using the instructions on Prof. Summet’s website.

Working with `pymysql` module much like `sqlite3` module. Two differences:

- Connect using hostname, username, and password instead of just database file.
- `pymysql` module uses a ANSI C "format" paramstyle, not the question mark ("qmark") paramstyle that `sqlite3` uses

These differences are illustrated in this snippet:

```
import pymysql
db = pymysql.connect( host = ’sqlserver.example.com’,
      passwd = ’SECRET12345’, user = ’dbuser’, db=’myDatabase’)
cursor = db.cursor()
sql = "INSERT INTO people (name, email, age) VALUES (%s, %s, %s)"
cursor.execute( sql, ( "Jay", "jay@example.com", ’77’) )
cursor.close()
db.commit()   #Makes sure the DB saves your changes!
db.close()
```