Server-Side Languages

Todd Smith
tbsmith@fullsail.com

Welcome to SSL Day 4!

Models and Database Connections

Day 4

Server-Side Languages

First, set up a database

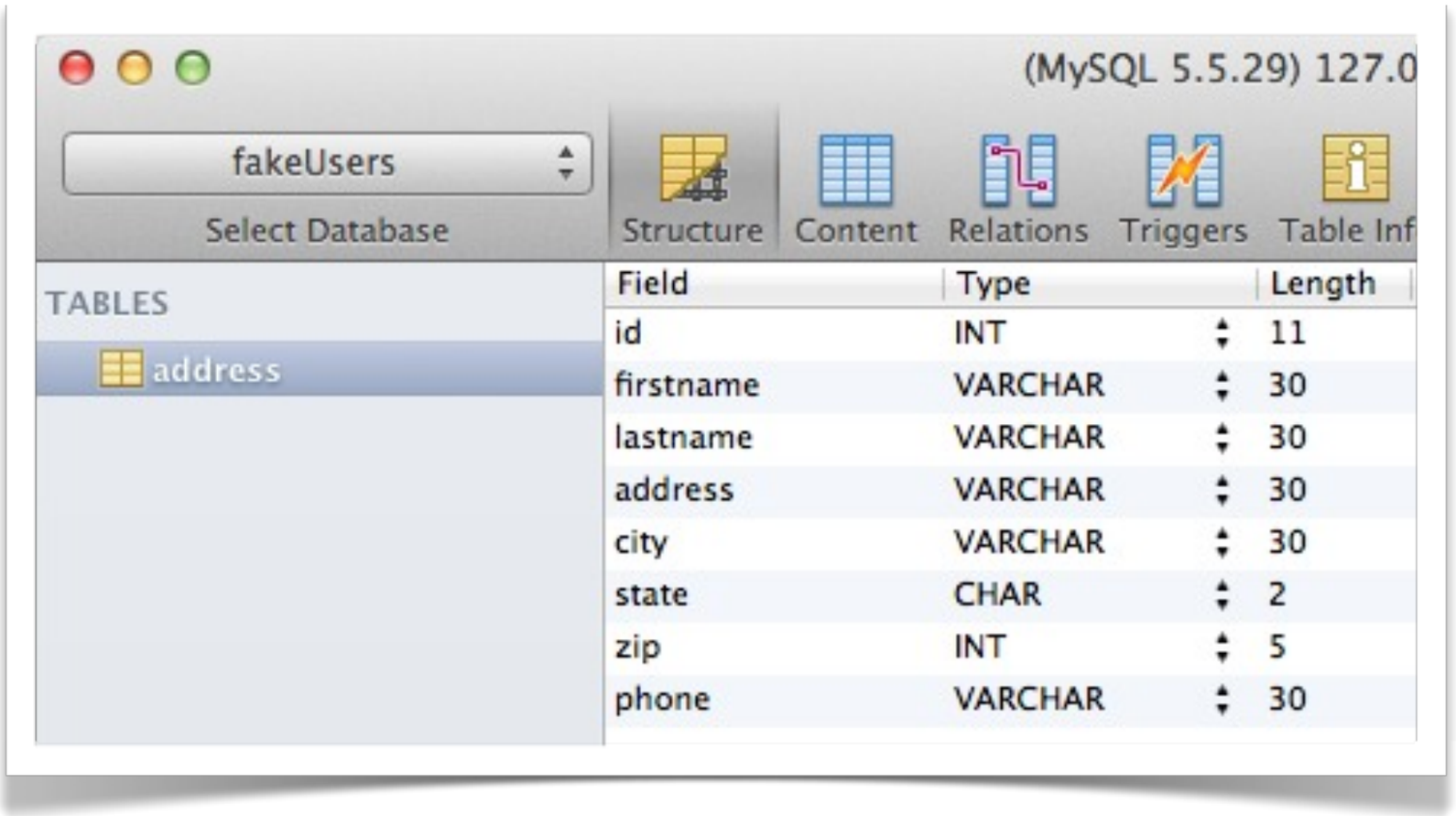
These examples will use a MySQL database.

You should set up a new database using the techniques you learned in DBS.

Day 4

Server-Side Languages

In these examples, the database is called “fakeUsers”.



Day 4

PHP DB Connection Model

The constructor function

```php
<?php

class DBConnector {

    private $db;

    function __construct() {
        $host = '127.0.0.1';
        $user = 'root';
        $pass = 'root';
        $port = '3306';
        $dbname = 'fakeUsers';
        $this->db = new PDO("mysql:host=$host;
                             port=$port;
                             dbname=$dbname",
                             $user, $pass);

    }
```

Using the PDO class, we can prevent SQL injection attacks.

The port 3306 is Apache's default MySQL port.

If you're using MAMP, your port is probably 8889.

Server-Side Languages

PHP DB Connection Model

Adding an entry

```
public function addUser($firstname='', $lastname='',
    $address='', $city='', $state='', $zip='', $phone='') {
        $stmnt = $this->db->prepare("insert into address
            (firstname, lastname, address, city, state, zip, phone)
            values (:firstname, :lastname, :address, :city, :state,
                :zip, :phone)");
        $stmnt->execute(array(
            ':firstname' => $firstname,
            ':lastname' => $lastname,
            ':address' => $address,
            ':city' => $city,
            ':state' => $state,
            ':phone' => $phone,
            ':zip' => $zip));
}
```

Inserting the values into the sql statement in this way will prevent SQL injection attacks.

Day 4

Server-Side Languages

PHP DB Connection Model

Selecting a random entry

```
public function getRandomUser() {
    $stmnt = $this->db->query("select * from address
        order by rand() limit 1");
    return $stmnt->fetchAll(PDO::FETCH_ASSOC);
}
```

Server-Side Languages

PHP DB Controller

Including the model

```
include "models/DBConnector.php";
```

Connecting to the database

```
$db = new DBConnector();
```

Server-Side Languages

PHP DB Controller

Getting a random user

```
$result = $db->getRandomUser();
```

Adding a user

```
$db->addUser("Tom", "Jones", "123 Easy St.",
    "Orlando", "FL", "32825", "407-867-5309");
```

Day 4

Server-Side Languages

The directory structure

The index file would route a request to list all users to the user controller.

The user controller would include the database model and ask it for a list of users.

The user controller would call the correct view, and put the user information into it.

The user controller would then display the list of users to the web user.

Server-Side Languages

Python DB Connection Model

The constructor function

```
#!/usr/bin/python

import mysql.connector


class DBConnector():

    def __init__(self):
        self.db = mysql.connector.connect(host="127.0.0.1",
                                          port=3306,
                                          user="root",
                                          passwd="root",
                                          db="fakeUsers")
```

Install this package from: http://goo.gl/RZS0n

The port 3306 is Apache's default MySQL port.

If you're using MAMP, your port is probably 8889.

Server-Side Languages

Python DB Connection Model

Adding an entry

This will create self.db

```
def add_user(self, fname='', lname='', address='',
             city='', state='', phone='', zip=''):

    self.get_connection()

    sql = "insert into address (firstname, lastname, address,\
          city, state, zip, phone) values (%(fname)s, \
            %(lname)s, %(address)s, %(city)s, %(state)s,\
            %(zip)s, %(phone)s)"

    user_info = {
        'fname': fname,
        'lname': lname,
        'address': address,
        'city': city,
        'state': state,
        'zip': zip,
        'phone': phone
    }

    cursor = self.db.cursor()
    cursor.execute(sql, user_info)
    self.db.commit()
    cursor.close()
    self.db.close()
```

Usually, for example, when you're typing in the console window in Sequel Pro, a setting called "auto-commit" is turned on.

Otherwise, when you insert, update, or delete from a database, your changes aren't saved until you commit them.

Inserting the values into the sql statement in this way will prevent SQL injection attacks.

Server-Side Languages

Python DB Connection Model

Getting a random entry

```
def get_random_user(self):
    sql = "select firstname, lastname, city, state,\
           zip, phone from address order by rand() limit 1"

    cursor = self.db.cursor()
    cursor.execute(sql)
    for firstname, lastname, city, state, zip, phone in cursor:
        print("{} {} lives in {}, {} {} with phone number {}".format(
            firstname, lastname, city, state, zip, phone))
    cursor.close()
    self.db.close()
```

Python DB Controller

Make sure you put the empty __init__.py file in the models directory!

Including the model

```
from models.DBConnector import DBConnector
```

Connecting to the database

```
$db = new DBConnector();
```