

INF1343, Winter 2012

# Data Modeling and Database Design

Yuri Takhteyev

Faculty of Information  
University of Toronto



This presentation is licensed under Creative Commons Attribution License, v. 3.0. To view a copy of this license, visit <http://creativecommons.org/licenses/by/3.0/>. This presentation incorporates images from the Crystal Clear icon collection by Everaldo Coelho, available under LGPL from <http://everaldo.com/crystal/>.

**Week 8**

**Building  
Database-Driven  
Web Applications**

# Interactivity

```
template = "Greetings, %s from %s!"
query_template = """select * from persona
where name="%s";"""
name = raw_input("What is your name? ")
query = fill_template(query_template, name)
execute_query(query)
for row in fetch_all_rows():
    world = get_row_value(row, "homeworld")
    if world==None:
        print greeting + name
    else:
        print fill_template(template,
                               name, world)
```

(We stopped here in class.)

# Or Rather

```
template = "Greetings, %s from %s!"
query_template = """select * from persona
where name="%s";"""
name = raw_input("What is your name? ")
query = fill_template(query_template, name)
execute_query(query)

if get_row_count() > 0 :
    rows = fetch_all_rows()
    row = rows[0]
    world = get_row_value(row, "homeworld")
    if world==None:
        print greeting + name
    else:
        print fill_template(template, name, world)
else :
    print "No such user!"
```

# Worlds as Passwords

```
template = "Greetings, %s from %s!"
query_template = """select * from persona
where name="%s";"""
name = raw_input("What is your name? ")
world = raw_input("Where are you from? ")
query = fill_template(query_template, name)
execute_query(query)

if get_row_count() > 0 :
    rows = fetch_all_rows()
    row = rows[0]
    real_world = get_row_value(row, "homeworld")
    if world==real_world:
        print fill_template(template, name, world)
    else:
        print "Access denied!"
else :
    print "No such user!"
```

# Worlds as Passwords

```
template = "Greetings, %s from %s!"
query_template = """select * from persona
where name="%s";"""
name = raw_input("What is your name? ")
world = raw_input("Where are you from? ")
query = fill_template(query_template, name)
execute_query(query)

if get_row_count() > 0 :
    rows = fetch_all_rows()
    row = rows[0]
    real_world = get_row_value(row, "homeworld")
    if world==real_world:
        print fill_template(template, name, world)
else :
    print "Access denied!"
```

# Worlds as Passwords

```
template = "Greetings, %s from %s!"
query_template = """select * from persona
where name="%s" and world="%s";"""
name = raw_input("What is your name? ")
world = raw_input("Where are you from? ")
query = fill_template(query_template, name, world)
execute_query(query)

if get_row_count() > 0 :
    print fill_template(template, name, world)

else :
    print "Access denied!"
```

# What about Inserts?

```
from padawan import *
connect_to_db("okenobi", "okenobi")
query_template = """insert into persona(name)
values ("%s");"""
name = raw_input("What is your name? ")
query = fill_template(query_template, name)
execute_query(query)

print "You are now in our database!"
```



# '%' in Queries

```
from padawan import *
connect_to_db("okenobi", "okenobi")
query_template = """select * from persona
where name like "%%s%%";"""
name = raw_input("What is your name? ")
query = fill_template(query_template, name)
print query
```

Use %% in the template when you want to have % in the query:

like "%s%" → like "%Bob%"

Sign Up

MySQL at Facebook is on Facebook

Sign up for Facebook to connect with MySQL at Facebook.



Devoted to making MySQL better. Written by people who work at Facebook.

### Information

Founded:  
2009

42,703 People Like This



Nesrine  
Messai



Krishna  
Chaitanya



Martin  
Prayoga

MySQL at Facebook



Wall

Info

Notes

Discussions

Links

MySQL at Facebook + Others

Just MySQL at Facebook

Just Others



**MySQL at Facebook** I just registered for PgWest 2010. I look forward to learning more about current work on PostgreSQL.



**PgWest 2010 | The PostgreSQL Conference**  
[www.postgresqlconference.org](http://www.postgresqlconference.org)

Thursday at 12:18pm · Comment · Like

Paul Saab, Bill Blum, Chetan Tanna and 51 others like this.

View all 7 comments



**MySQL at Facebook** I think they both have awesome technology and continue to improve.  
Yesterday at 9:24am · 2 people · Flag



**Ricky Elrod** Indeed they do, I was just pulling your leg. Have fun at the conference :)  
Yesterday at 9:26am · Flag



**MySQL at Facebook** Kristian Nielsen had a very interesting presentation on making replication better in MariaDB. Perhaps one day he will share that (hint, hint).

October 11 at 8:21am · Comment · Like

Kian Sin Teo, Donald G. Lane Jr., Roja Rani Penmetsa and 38 others like this.

# Telnet (1969)



"r2", "alice", "pumpkins"

"alice", "pumpkins"

ok

"alice@r2:~\$"

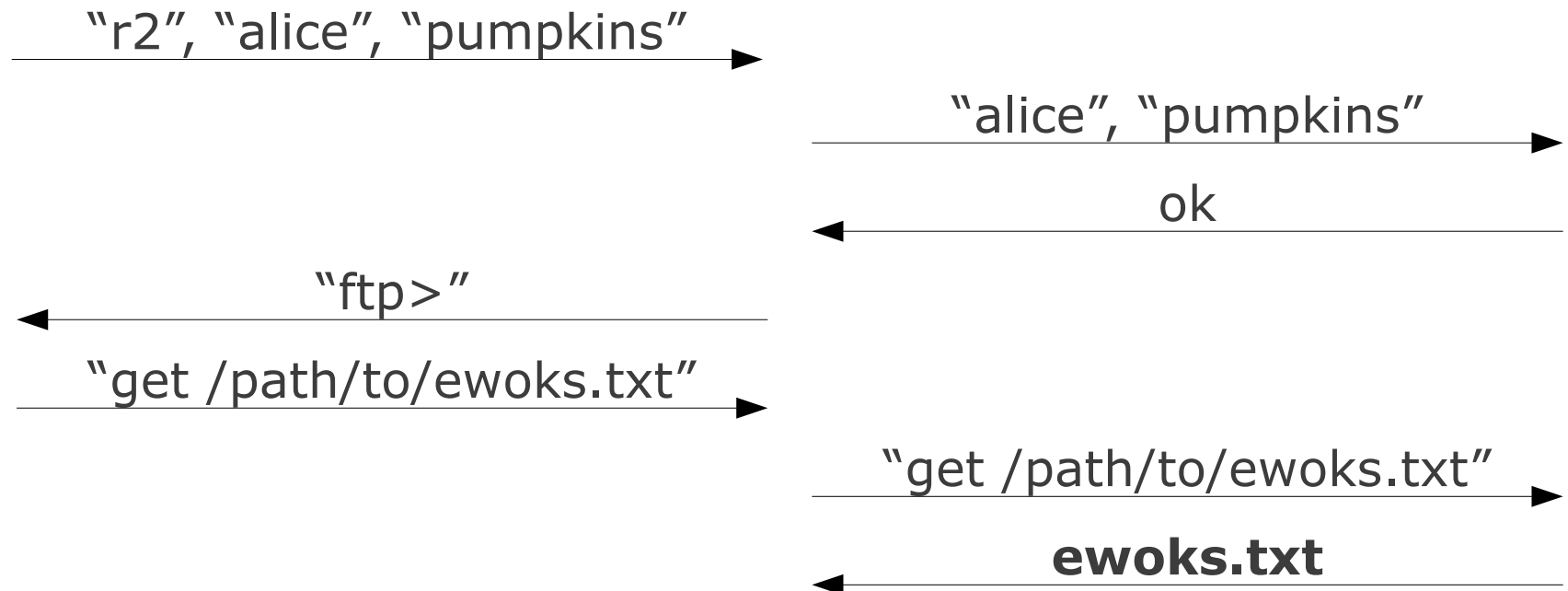
"python greetings.py"

"python greetings.py"

"What is your name?"

"What is your name?"

# FTP (1971)



the file is saved locally

# Anonymous FTP



"r2", "/path/to/ewoks.txt"

"get /path/to/ewoks.txt"

**ewoks.txt**

the file is saved locally

ftp://yoda/path/to/ewoks.txt

# HTTP (1991)



"r2", "/path/to/ewoks.html" →

"GET /path/to/ewoks.html" →

← the **content** of ewoks.html

← ewoks.html is displayed

http://yoda/path/to/ewoks.html

# HTML

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>
      Ewoks
    </title>
  </head>
  <body>
    <h1>
      The List of Ewoks
    </h1>
    ...
```

See: <http://www.w3schools.com/html/default.asp>

# Basic Web Setup

## **1. Create an HTML file.**

Call it "test.html".

## **2. Put it in ~/public\_html.**

Use an SCP client.

~ = your home directory.

## **3. Set permissions.**

Make the file readable by "other".

## **4. Access it in the browser.**

<http://r2.ischool.utoronto.ca/~okenobi/test.html>

(Replace "okenobi" with your username.)



# Generated HTML

```
from padawan import *
connect_to_db("starwars", "okenobi")
query = "select * from persona"
execute_query(query)
html_template = """
<p><b>%s</b> is from <b>%s</b></p>
"""
for row in fetch_all_rows():
    name = get_row_value(row, "name")
    world = get_row_value(row, "homeworld")
    if world!=None:
        print fill_template(html_template,
                            name, world)
```

# Run with Redirection

```
python personas.py > ~/public_html/personas.html
```

(You can use WinSCP to check that the file ended up in the right place and that the permissions are correct.)

# CGI



`"r2", "/path/to/ewoks.py"`

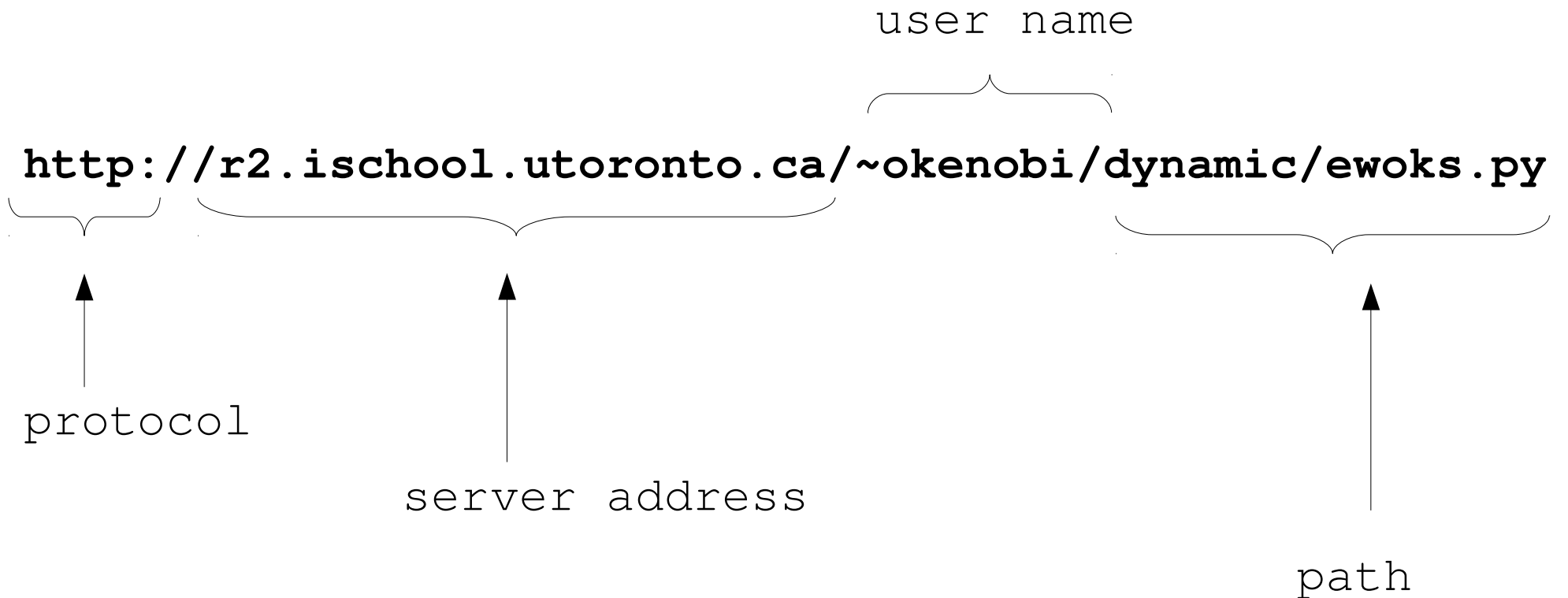
`"GET /path/to/ewoks.cgi"`

**the html output of ewoks.py**

**html output** is displayed

ewoks.py contains **code**, but **returns** HTML

# personas.py



# personas.py

```
#!/usr/bin/python  
  
print "Content-Type: text/html"  
print  
print "<h1>Personas</h1>"
```

Again, create locally, copy to r2 with WinSCP.  
The file should be in `~/public_html/dynamic`.

Set permissions to allow "other" and "group"  
to **read** and **execute** but *not* to **write**.

# CGI with Padawan

```
#!/usr/bin/padawan
print "Content-Type: text/html; encoding=utf8"
print
connect_to_db("starwars", "okenobi")
query = "select * from persona"
execute_query(query)
html_template = """
<p><b>%s</b> is from <b>%s</b></p>
"""
for row in fetch_all_rows():
    name = get_row_value(row, "name")
    world = get_row_value(row, "homeworld")
    if world!=None:
        print fill_template(html_template,
                            name, world)
```

# Passing Parameters

## Option 1, "GET" (in the URL)

<http://r2.ischool.utoronto.ca/~okenobi/dynamic/personas.py?species=Human>

`.../personas.py?species=Human`

## Option 2, "POST" (invisibly)

(we'll get back to this)

# Accessing Parameters

```
print_form()
```

```
if form_has_field("species") :  
    print "Species is defined!"
```

```
else :  
    print "Species is not defined. : ("
```

```
s = get_form_field_value("species")  
print "The species is " + s
```



# The Wrong Way

```
template = """
select * from persona
where species='%s';
"""
```

Wrong!!!

```
if form_has_field("species") :
    s = get_form_field_value("species")
    query = fill_template(template, s)
    print query
    execute_query(query);
    count = get_row_count()
    print "Got ", count, "results!"
```

# Here Is Why

[http://r2.ischool.utoronto.ca/~okenobi/dynamic/personas.py?](http://r2.ischool.utoronto.ca/~okenobi/dynamic/personas.py?species=Human%3Binsert+into+persona+(name,species)+values+('X','Human')%3Bselect+*+from+persona+where+)

[species=Human%3Binsert+into+persona+\(name,species\)+values+\('X','Human'\)%3Bselect+\\*+from+persona+where+](http://r2.ischool.utoronto.ca/~okenobi/dynamic/personas.py?species=Human%3Binsert+into+persona+(name,species)+values+('X','Human')%3Bselect+*+from+persona+where+)

# Right

```
template = """
select * from persona
where species='%s';
"""

if form_has_field("species") :
    s = get_form_field_value("species")
    safe_s = escape_string(s)
    query = fill_template(template,
                          safe_s)

    print query
    execute_query(query);
    count = get_row_count()
    print "Got ", count, "results!"
```

# HTML Forms

```
<form action="/~okenobi/dynamic/personas.py"
      method="get">
  Please select a species:
  <input type="text" name="species" />
  <input type="submit" />
</form>
```

# HTML Forms

```
<form action="/~okenobi/dynamic/personas.py"
      method="get">
  Please select a species:
  <select name="species">
    <option value="Human">Human</option>
    <option value="Ewok">Ewok</option>
    <option value="Wookiee">Wookiee</option>
  </select>
  <input type="submit" />
</form>
```

# HTML Forms: POST

```
<form action="/~kenobio1/cgi-bin/personas3.cgi"
      method="post">
  Please select a species:
  <select name="species">
    <option value="Human">Human</option>
    <option value="Ewok">Ewok</option>
    <option value="Wookiee">Wookiee</option>
  </select>
  <input type="submit" />
</form>
```



initial request (a URL)

an HTML page with a form

a new request (another URL)

an HTML page with a form

a request with parameters

an HTML page with hidden form elements

a request with parameters

an HTML page with hidden form elements



# Prefilled Forms

Please select a species:

```
<input type="text" name="species"  
  value="Human" />
```



# Hidden Elements

Please select a species:

```
<input type="hidden" name="species"  
value="Human" />
```

# Or Just in Links

```
<a href="dynamic/personas.py?species=Human">  
Click here for the Humans  
</a>
```

Questions?