

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 7

Embedded SQL

Facebook runs on MySQL

facebook

Email

Password

Login

Keep me logged in

[Forgot your password?](#)

Sign Up

MySQL at Facebook is on Facebook

Sign up for Facebook to connect with MySQL at Facebook.



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

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.

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

Information

Founded:
2009

42,703 People Like This



Nesrine
Messai



Krishna
Chaitanya

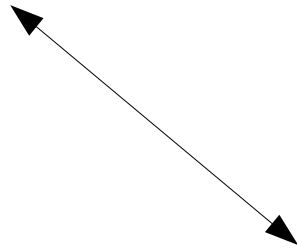


Martin
Prayoga





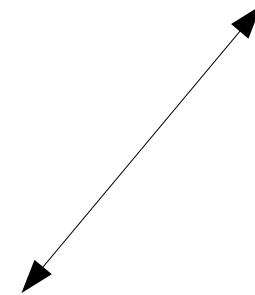
Alice



application software



Bob



database

"persistent storage"



Computer Languages

Domain Specific:

Has a narrow purpose.

Examples: SQL, HTML, Matlab.

General Purpose:

Can be used for anything.

Examples: Python, Java, PHP.

Computer Languages

Declarative:

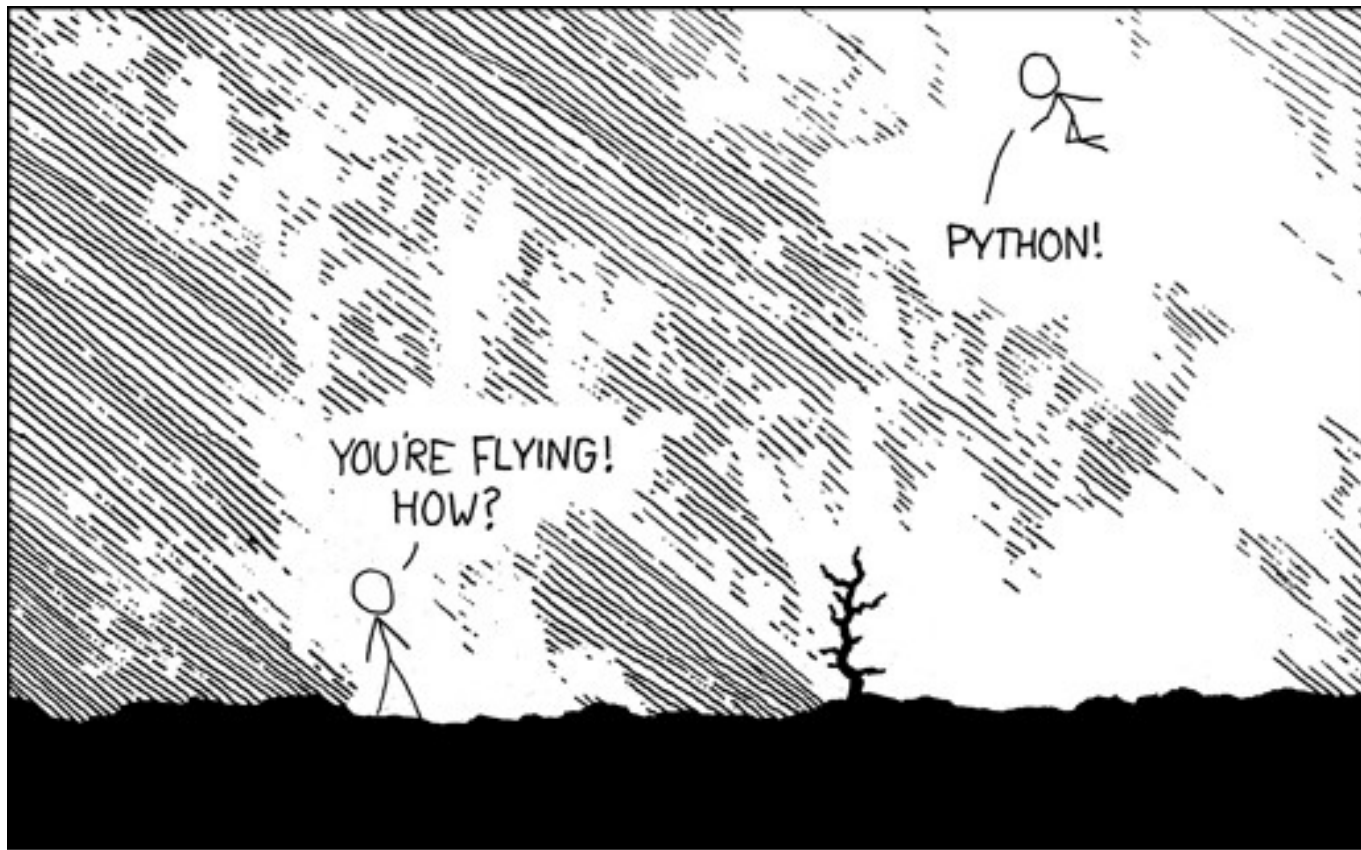
“Tell me what you want.”

Examples: SQL, HTML, ~~Matlab~~.

Procedural:

“Tell me what you want me to do.”

Examples: Python, Java, PHP.



I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST
`print "Hello, world!"`

I DUNNO...
DYNAMIC TYPING?
WHITESPACE?


COME JOIN US!
PROGRAMMING IS FUN AGAIN!
IT'S A WHOLE NEW WORLD
UP HERE!



BUT HOW ARE YOU FLYING?

I JUST TYPED
`import antigravity`

THAT'S IT?



... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

Python

```
print "Hello, World!"
```

```
greeting = "Hello, World!"  
print greeting
```


Python on R2 via SSH

On Windows: use PuTTY

Hostname: r2.ischool.utoronto.ca

Username: same as for MySQL WB

Password: same as for MySQL WB

On OSX/Unix: use Terminal App

```
ssh «username»@r2.ischool.utoronto.ca
```

```
ssh okenobi@r2.ischool.utoronto.ca
```

Once connected, type "python".

A Few Tips

Ctrl+D to quit.

Ctrl+L to clear screen.

Up arrow to bring back earlier commands.

Trying Some Python

```
print "Hello, World!"
```

```
greeting = "Hello, World!"  
print greeting
```

```
x = 2  
print x  
x = x + 2  
print x
```

Adding Strings

```
greeting = "Hello, "  
print greeting + "Bob" + "!"  
  
name = "Alice"  
print greeting + name + "!"  
  
name = "Ze"  
print greeting + name + "!"
```

Lists

```
names = ["Alice", "Bob", "Ze"]  
print names[1]  
  
greeting = "Hello, "  
print greeting + names[1]  
  
i = 2  
print greeting + names[i]
```

Oops

```
i = 4  
print greeting + names[i]
```

Functions

```
print len(names)
```

```
print max(1, 22, 19, 2)
```

Conditions

```
names = ["Alice", "Bob", "Ze"]  
i = 2  
if i < len(names):  
    print greeting + names[i]  
else:  
    print "i is too large!"
```



4 spaces. Do **not** use tabs.

A Programming Editor

On Windows: Notepad++

On OSX: TextWranger

Getting the Files to R2

On Windows: WinSCP

On OSX: Cyberduck

Same hostname, username,
password as for SSH.

Executing a Python File

```
python «filename»
```

For instance:

```
python greetings.py
```

Loops

```
names = ["Alice", "Bob", "Ze"]  
for name in names:  
    print greeting + name
```



4 spaces. Do **not** use tabs.

Combining

```
names = ["Alice", "Bob", "Ze"]
for name in names:
    if name=="Ze":
        print "Oi, Ze"
    else:
        print greeting+name
```



4 spaces. Do **not** use tabs.

Using the DB

```
from padawan import *  
connect_to_db("starwars", "okenobi")  
query = "select * from persona"  
execute_query(query)  
print get_row_count()
```

Padawan gives you a subset of the DB functionality. For more options you can try Python's module MySQLdb (this is what Padawan uses behind the scenes).

Getting Values

```
connect_to_db("starwars", "okenobi")
query = "select * from persona"
execute_query(query)
print get_row_count()
```

```
rows = fetch_all_rows()
row = rows[0]
print row
# or: row = get_row(0)
print row[2]
print get_row_value(row, "species")
```

Loops

```
connect_to_db("starwars", "okenobi")
query = "select * from persona"
execute_query(query)
print get_row_count()

rows = fetch_all_rows()
for row in rows :
    print get_row_value(row, "species")
```


Longer Loops

```
connect_to_db("starwars", "okenobi")
query = "select * from persona"
execute_query(query)
print get_row_count()

for row in fetch_all_rows() :
    name = get_row_value(row, "name")
    world = get_row_value(row, "homeworld")
    print greeting + name, world
print "done!"
```

Longer Loops

```
connect_to_db("starwars", "okenobi")
query = "select * from persona"
execute_query(query)
print get_row_count()

for row in fetch_all_rows() :
    name = get_row_value(row, "name")
    world = get_row_value(row, "homeworld")
    print greeting + name + world
print "done!"
```

TypeError: cannot concatenate 'str' and 'NoneType' objects

Yet Longer Loops

```
connect_to_db("starwars", "okenobi")
query = "select * from persona"
execute_query(query)
print get_row_count()

for row in fetch_all_rows():
    name = get_row_value(row, "name")
    world = get_row_value(row, "homeworld")
    if world==None:
        print greeting + name
    else:
        print greeting+name+" from "+world
```

Templates

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

(Alternatively, use Python's "%" operator.)

Fancier Queries

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

Query Templates

```
template = "Greetings, %s from %s!"
query_template = """select * from persona
where name="%s";"""
query = fill_template(query_template,
                      "Jabba")
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)
```

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!"
```

Questions?