Building Database Apps with Go

Me

- Cofounder of @VividCortex
- Author of High Performance MySQL
- Get in touch!
  - @xaprb on Twitter
  - baron@vividcortex.com
  - http://www.linkedin.com/in/xaprb

Go Database Design Patterns

Things To Know

- What is database/sql?
- Where is the documentation?
- How do I create a database connection?
- How do I query a connection?
- How do I retrieve results from a query?
- What are the common operations?

Overview of database/sql

- A generic, minimal interface for SQL-like databases
- Verbs and nouns:
  - Open, Prepare, Exec, Query, QueryRow, Scan, Close
  - DB, Stmt, Value, Row, Rows, Result
  - Tx, Begin, Commit, Rollback
- Docs: http://golang.org/pkg/database/sql/
- Tutorial: https://github.com/VividCortex/go-database-sql-tutorial

```
package main

import (
    _ "github.com/go-sql-driver/mysql"

    "database/sql"
    "log"
)

func main() {
    db, err := sql.Open("mysql",
        "user:password@tcp(127.0.0.1:3306)/test")
    if err != nil {
        log.Println(err)
    }
    defer db.Close()
}
```

One-Shot Query

```
var str string
q := "select hello from hello.world where id = 1"
err = db.QueryRow(q).Scan(&str)
if err != nil {
   log.Println(err)
}
log.Println(str)
```

Query...

```
   var id int
   var str string
   q := "select id, hello from hello.world where id
= ?"

   rows, err := db.Query(q, 1)
   if err != nil {
      log.Fatal(err)
   }
   defer rows.Close()
```

... Then Fetch Rows

```
for rows.Next() {
    err = rows.Scan(&id, &str)
    if err != nil {
        log.Fatal(err)
    }
    // Use the variables scanned from the row
}
err = rows.Err()
if err != nil {
    rows.Close()
    log.Fatal(err)
}
```

Prepare...

```
stmt, err := db.Prepare(
    "select id, hello from hello.world
     where id = ?")
if err != nil {
    log.Fatal(err)
}
defer stmt.Close()
```

... Then Execute & Fetch

```
rows, err := stmt.Query(1)
if err != nil {
    log.Fatal(err)
}
defer rows.Close()


for rows.Next() {
    // ...
}
```

Insert...

```
stmt, err := db.Prepare(
    "insert into hello.world(hello) values(?)")
if err != nil {
    log.Fatal(err)
}
defer stmt.Close()

res, err := stmt.Exec("hello, Dolly")
if err != nil {
    log.Fatal(err)
}
```

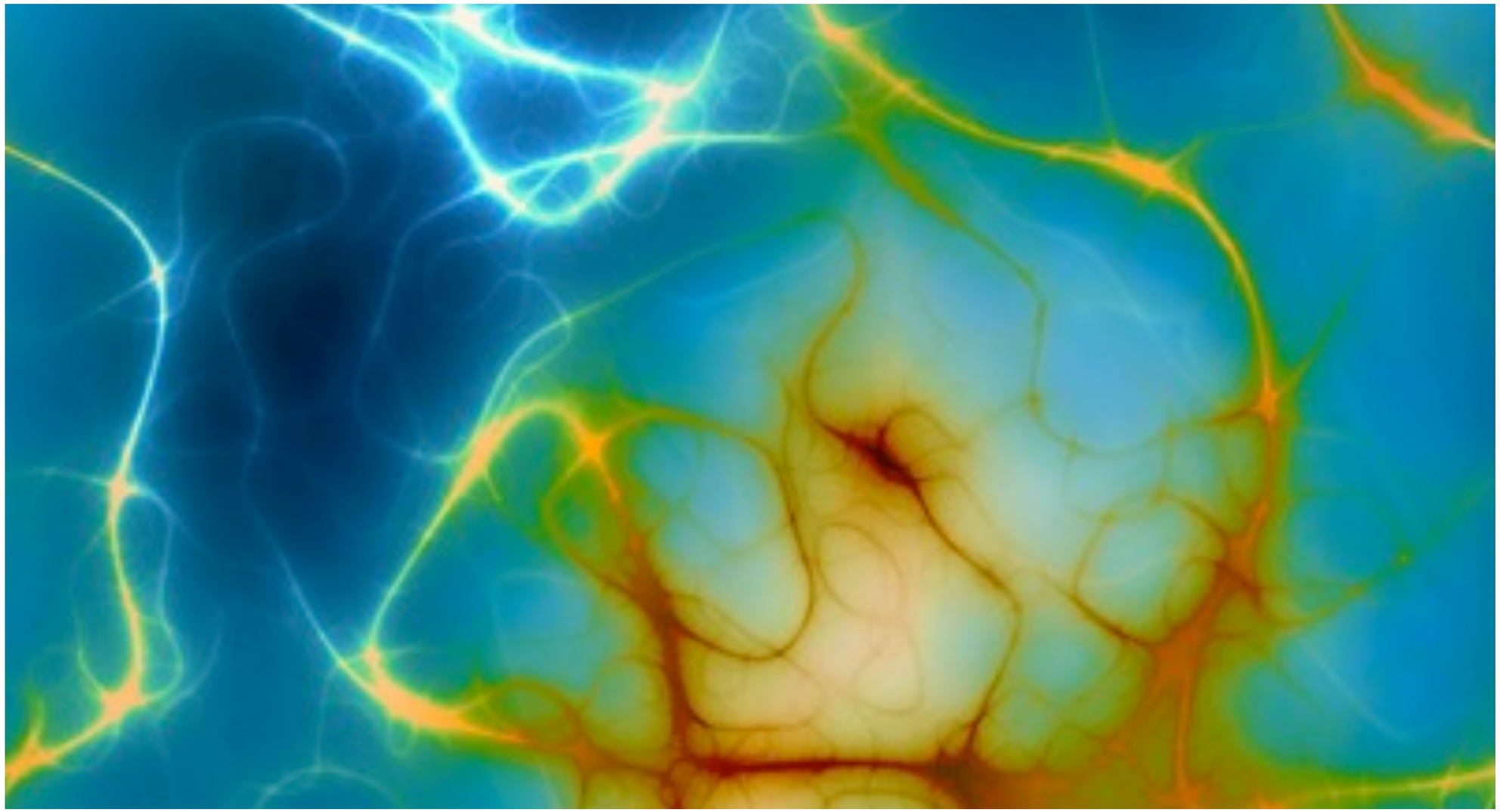
... And Check Results

```
lastId, err := res.LastInsertId()
if err != nil {
    log.Fatal(err)
}
rowCnt, err := res.RowsAffected()
if err != nil {
    log.Fatal(err)
}
```

Docs: “Pleasantly Lightweight!”

Connection Pooling

Big Unsigned Integers

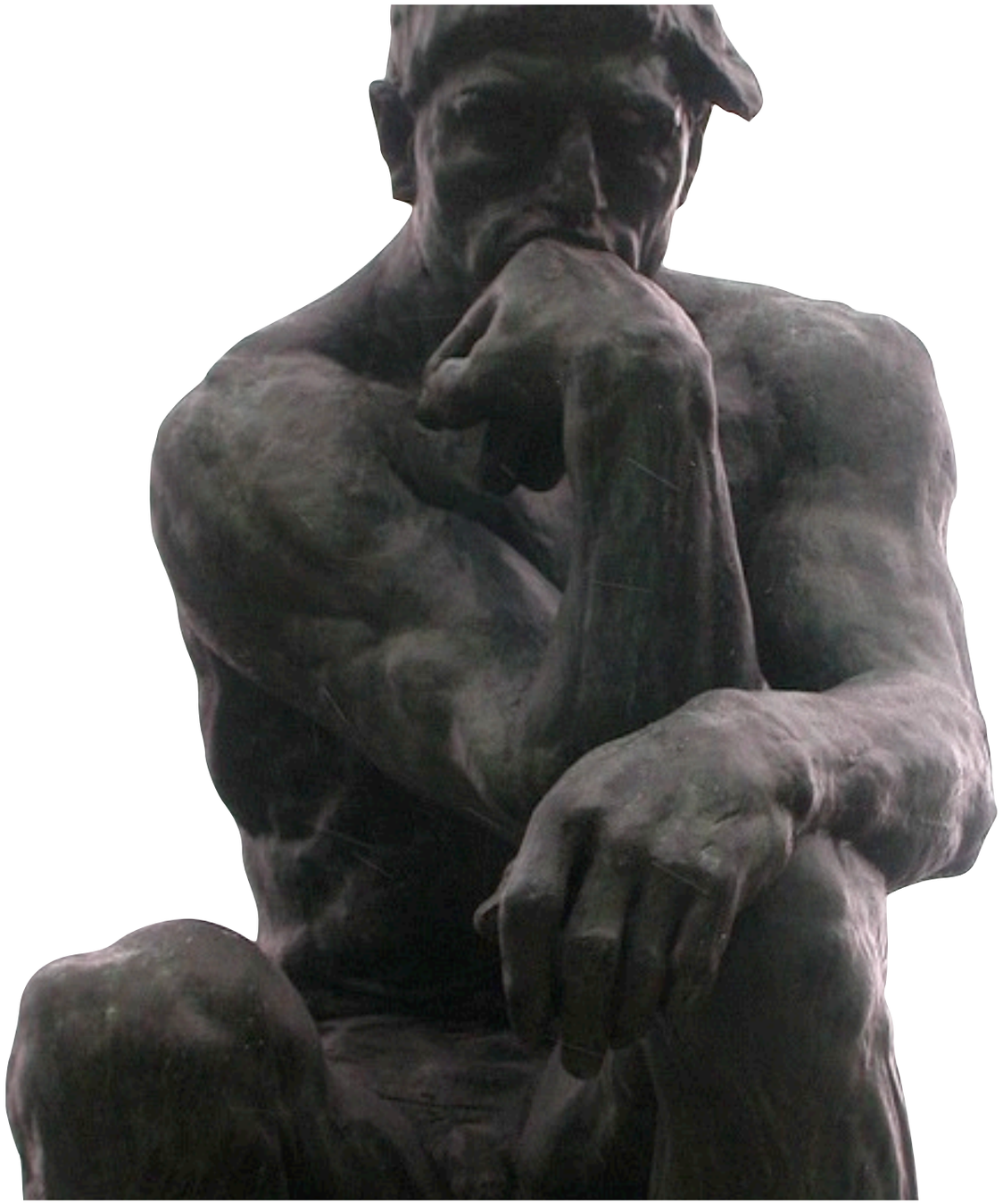


Unexpected Connections

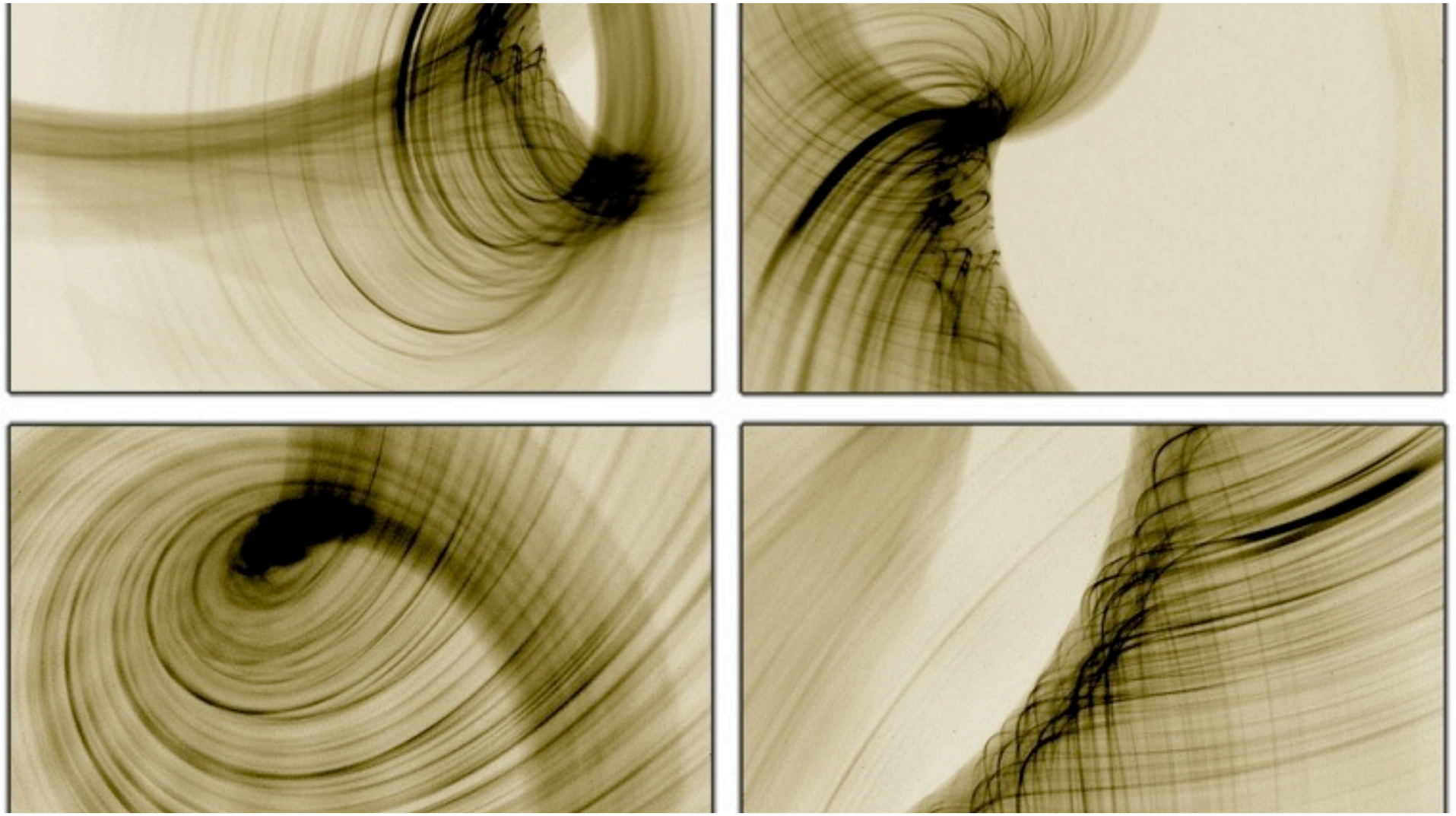
Resource Exhaustion

```
// Don’t do this!
for i := 0; i < 50; i++ {
   _, err := db.Query("DELETE FROM hello.world")
}


// Use this instead!
for i := 0; i < 50; i++ {
   _, err := db.Exec("DELETE FROM hello.world")
}
```



Type Introspection



Handling NULLs

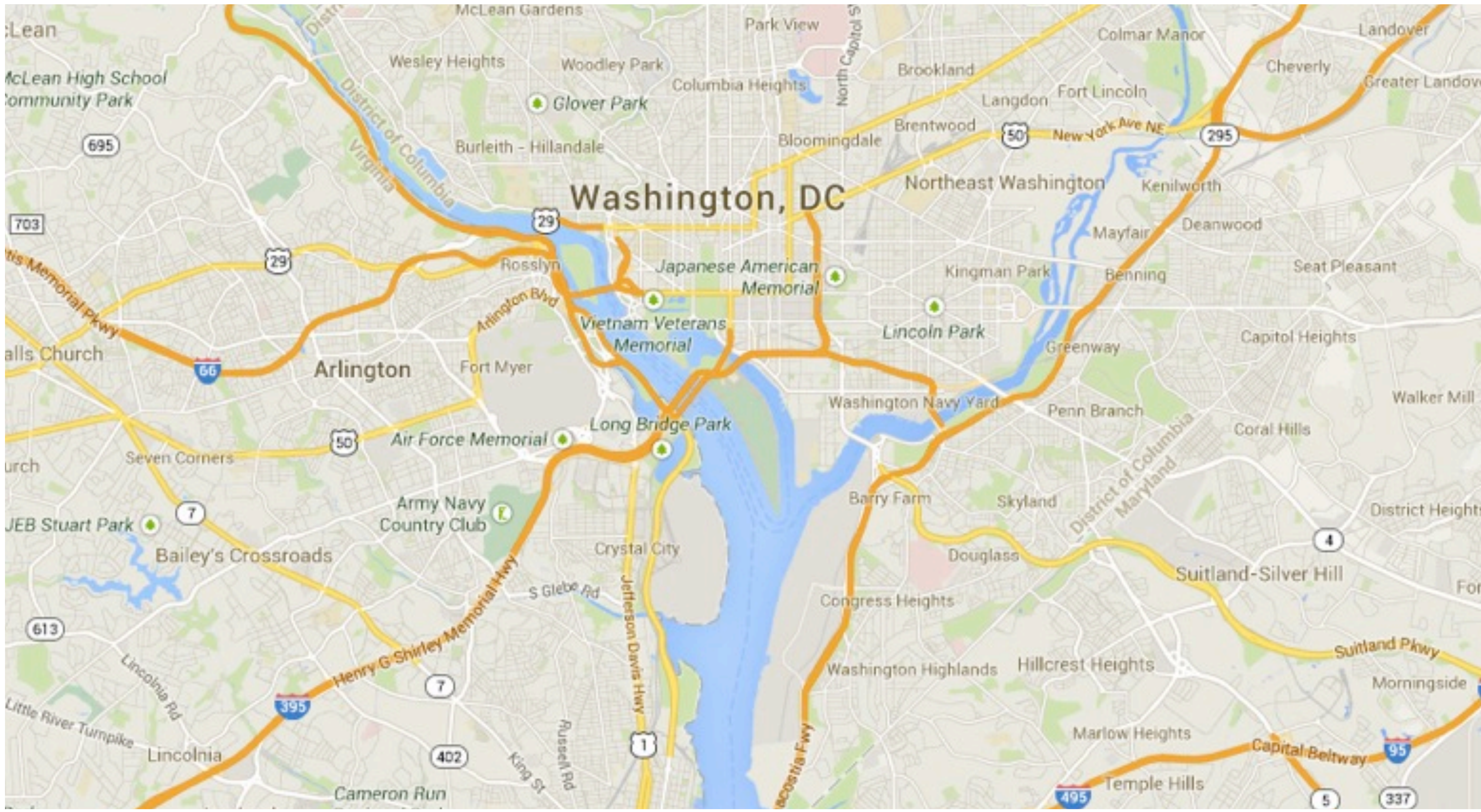
Working with NULL

```
var s NullString
err := db.QueryRow(
   "SELECT name FROM foo WHERE id=?",
   id).Scan(&s)

if s.Valid {
   // use s.String
} else {
   // NULL value
}
```

NULL Makes Code Ugly

- Boilerplate code everywhere
- There's no sql.NullUint64
- Nullability is tricky, not future-proof
- It's not very Go-ish (no useful default zero-value)



McLean Gardens
Park View
Colmar Manor
Landover
McLean High School Community Park
Wesley Heights
Woodley Park
Columbia Heights
Brookland
Cheverly
Greater Landover
Glover Park
Langdon
Fort Lincoln
Burleith - Hillandale
Bloomingdale
Brentwood
New York Ave NE
Washington, DC
Northeast Washington
Kenilworth
Mayfair
Deanwood
Rosslyn
Japanese American Memorial
Kingman Park
Benning
Seat Pleasant
Arlington Blvd
Vietnam Veterans Memorial
Lincoln Park
Capitol Heights
Falls Church
Arlington
Fort Myer
Greenway
Washington Navy Yard
Penn Branch
Walker Mill
Long Bridge Park
Coral Hills
Air Force Memorial
Seven Corners
Barry Farm
Skyland
District of Columbia
Maryland
District Heights
Army Navy Country Club
JEB Stuart Park
Crystal City
Bailey's Crossroads
Douglass
Suitland-Silver Hill
S Glebe Rd
Congress Heights
Jefferson Davis Hwy
Henry G Shirley Memorial Hwy
Suitland Pkwy
Washington Highlands
Hillcrest Heights
Morningside
Lincolnia Rd
Little River Turnpike
Lincolnia
King St
Russell Rd
Marlow Heights
Capital Beltway
Temple Hills
Cameron Run

ORMs

Resources

- http://golang.org/pkg/database/sql/
- https://github.com/VividCortex/go-database-sql-tutorial
- https://github.com/go-sql-driver/mysql
- http://jmoiron.net/blog/

Questions?

baron@vividcortex.com

@xaprb • linkedin.com/in/xaprb

VividCortex

Image Credits

- http://www.flickr.com/photos/simens/6306917636/
- http://www.flickr.com/photos/dexxus/5794905716/
- http://www.flickr.com/photos/sebastian_bergmann/202396633/
- http://www.flickr.com/photos/doug88888/4794114114/
- http://www.flickr.com/photos/oatsy40/6443878013/
- http://www.sxc.hu/photo/1160562/
- Google Maps (screenshot)
- http://www.flickr.com/photos/paperpariah/4150220583/
- http://www.flickr.com/photos/zooboing/4743616313/
- http://www.flickr.com/photos/dseneste/5912382808/
- http://www.flickr.com/photos/clickykbd/66165381/sizes/l/
- http://www.flickr.com/photos/mamnaimie/5576980406/