Sane SQL Change Management with Sqitch

David E. Wheeler
http://sqitch.org/

PDX.pm
January 9, 2013

Text: Attribution-Noncommercial-Share Alike 3.0 United States:
http://creativecommons.org/licenses/by-nc-sa/3.0/us/
Images licensed independently and © Their respective owners.

iovation

Whats Wrong with Migrations?

Whats Wrong with Migrations?

- Incomplete mini-language

Whats Wrong with Migrations?

- Incomplete mini-language
- No logical replication integration

Whats Wrong with Migrations?

- Incomplete mini-language
- No logical replication integration
- Numbered scripts hard to track

Whats Wrong with Migrations?

- Incomplete mini-language
- No logical replication integration
- Numbered scripts hard to track
- No VCS awareness

What about SQL Migrations?

- Incomplete mini-language
- No logical replication integration
- Numbered scripts hard to track
- No VCS awareness

What about SQL Migrations?

- ~~Incomplete mini-language~~
- No logical replication integration
- Numbered scripts hard to track
- No VCS awareness

What about SQL Migrations?

- ~~Incomplete mini-language~~
- ~~No logical replication integration~~
- Numbered scripts hard to track
- No VCS awareness

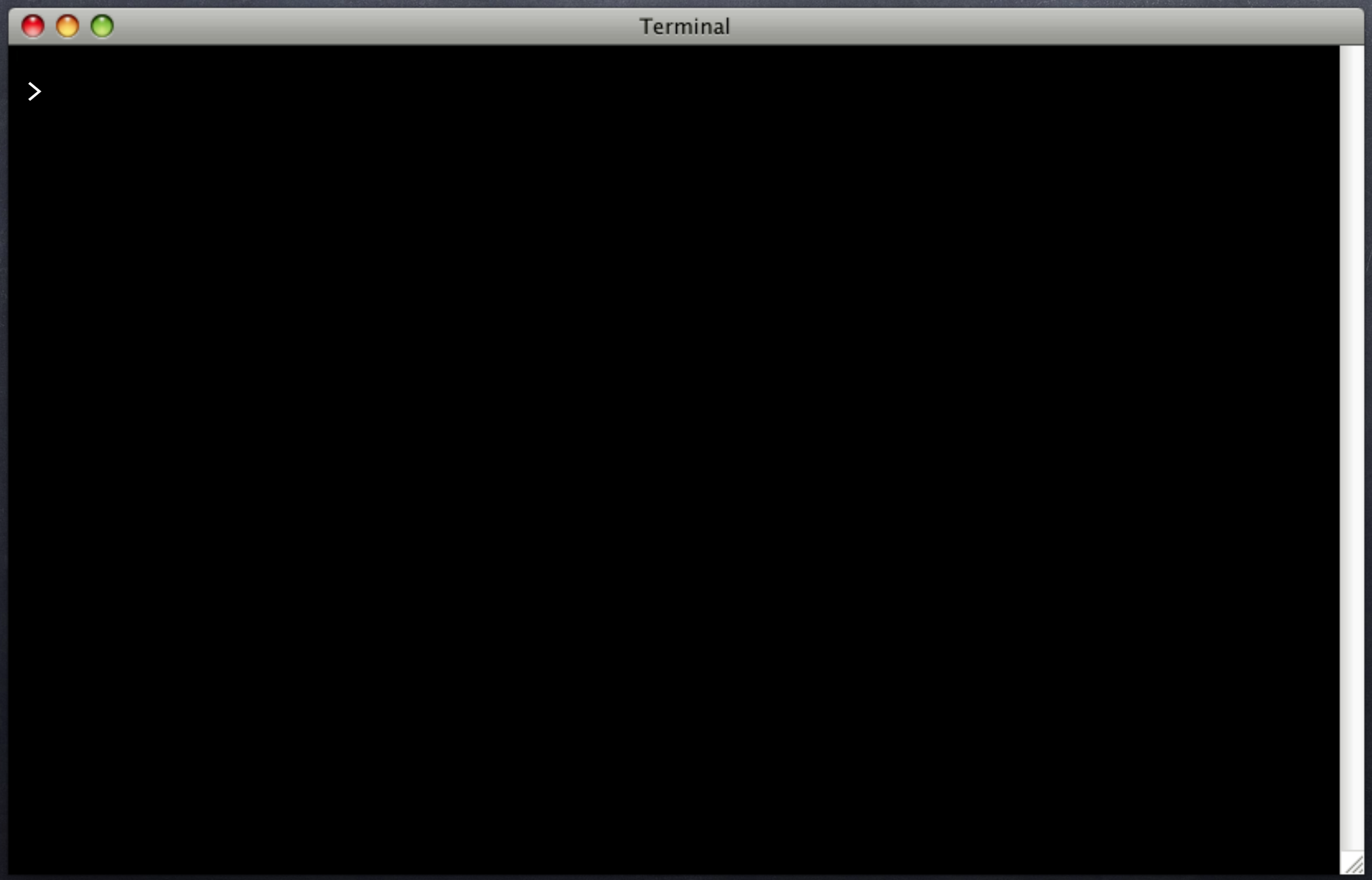
What about SQL Migrations?

- ~~Incomplete mini-language~~
- ~~No logical replication integration~~
- Numbered scripts hard to track
- No VCS awareness

What about SQL Migrations?

- ~~Incomplete mini-language~~
- ~~No logical replication integration~~
- Numbered scripts hard to track
- No VCS awareness
- Managing procedures is a PITA

Imagine this Change



Imagine this Change

Terminal

```
> git diff
diff --git a/deploy/recur.sql b/deploy/recur.sql
index 622d52e..56e419e 100644
--- a/deploy/recur.sql
+++ b/deploy/recur.sql
@@ -22,7 +22,10 @@
                    recurrence <> 'none'
                OR  (
                        recurrence = 'none'
-                   AND starts_at BETWEEN range_start AND range_end
+                   AND (
+                           starts_at BETWEEN range_start AND range_end
+                        OR ends_at   BETWEEN range_start AND range_end
+                   )
                )
            )
     LOOP
```

Imagine this Change

Terminal

```
> git diff
diff --git a/deploy/recur.sql b/deploy/recur.sql
index 622d52e..56e419e 100644
--- a/deploy/recur.sql
+++ b/deploy/recur.sql
@@ -22,7 +22,10 @@
                    recurrence <> 'none'
                OR  (
                        recurrence = 'none'
-                   AND starts_at BETWEEN range_start AND range_end
+                   AND (
+                           starts_at BETWEEN range_start AND range_end
+                        OR ends_at   BETWEEN range_start AND range_end
+                   )
                )
            )
    LOOP
```

Simple, right?

Not So Much

Not So Much

- Paste entire function to new “up” script

Not So Much

- Paste entire function to new “up” script
- Edit the new file

Not So Much

- Paste entire function to new “up” script
- Edit the new file
- Copy the function to the new “down” script

Not So Much

- Paste entire function to new “up” script
- Edit the new file
- Copy the function to the new “down” script
- Three copies of the function!

Not So Much

- Paste entire function to new “up” script
- Edit the new file
- Copy the function to the new “down” script
- Three copies of the function!
- No real source code management

Not So Much

- Paste entire function to new “up” script
- Edit the new file
- Copy the function to the new “down” script
- Three copies of the function!
- No real source code management
- This sucks

Introducing Sqitch

Sq—what?

sql changes

Sq—what?

sq ch

Sq—what?

sqitch

Sq—what?

sqitch

There is no “u”

Sqitch Philosophy

Sqitch Philosophy

- No opinions

Sqitch Philosophy

- No opinions
- Native scripting

Sqitch Philosophy

- No opinions
- Native scripting
- Cross-project dependency resolution

Sqitch Philosophy

- No opinions
- Native scripting
- Cross-project dependency resolution
- No numbering

Sqitch Philosophy

- No opinions
- Native scripting
- Cross-project dependency resolution
- No numbering
- Distribution bundling

Sqitch Philosophy

- No opinions
- Native scripting
- Cross-project dependency resolution
- No numbering
- Distribution bundling
- Integrated validation testing

Sqitch Philosophy

- No opinions
- Native scripting
- Cross-project dependency resolution
- No numbering
- Distribution bundling
- Integrated validation testing
- VCS integration

Sqitch Philosophy

Sqitch Philosophy

- Reduced duplication

Sqitch Philosophy

- Reduced duplication
- Built-in configuration

Sqitch Philosophy

- Reduced duplication
- Built-in configuration
- Iterative development

Sqitch Philosophy

- Reduced duplication
- Built-in configuration
- Iterative development
- Deployment planning

Sqitch Philosophy

- Reduced duplication
- Built-in configuration
- Iterative development
- Deployment planning
- Git-style interface

Sqitch Philosophy

- Reduced duplication
- Built-in configuration
- Iterative development
- Deployment planning
- Git-style interface
- Deployment tagging

Sqitch Terminology

Sqitch Terminology

- change

Sqitch Terminology

- change
- tag

Sqitch Terminology

- change
- tag
- state

Sqitch Terminology

- change
- tag
- state
- plan

Sqitch Terminology

- change
- tag
- state
- plan
- add

Sqitch Terminology

- change
- tag
- state
- plan
- add
- deploy

Sqitch Terminology

- change
- tag
- state
- plan
- add
- deploy
- revert

Sqitch Terminology

- change
- tag
- state
- plan
- add
- deploy
- revert
- rebase

Sqitch Terminology

- change
- tag
- state
- plan
- add
- deploy
- revert
- rebase
- rework

Sqitch Terminology

- change
- tag
- state
- plan
- add
- deploy
- revert
- rebase
- rework
- verify

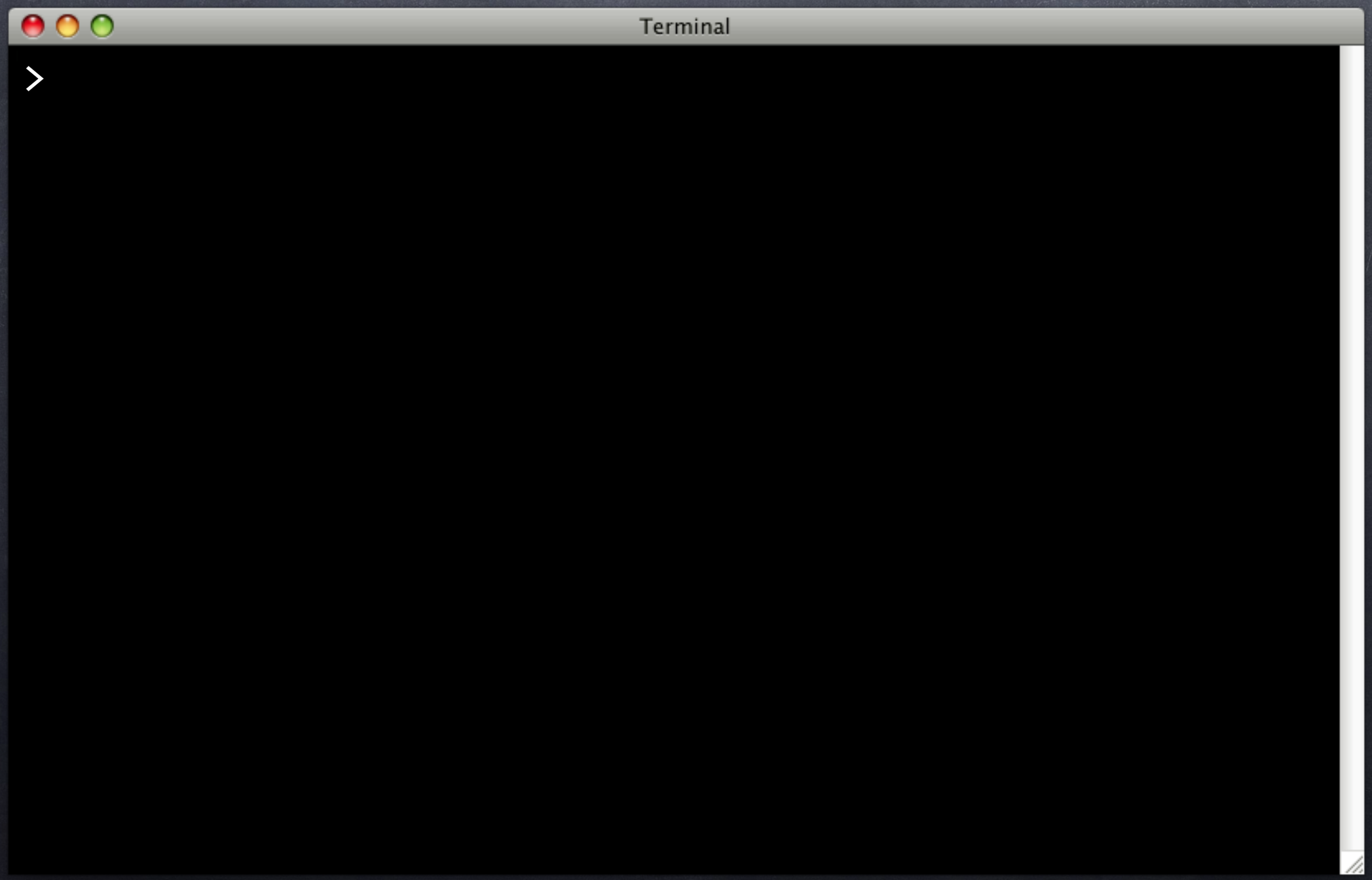
Sqitch Terminology

- change
- tag
- state
- plan
- add
- deploy
- revert
- rebase
- rework
- verify
- committer

Sqitch Terminology

- change
- tag
- state
- plan
- add
- deploy
- revert
- rebase
- rework
- verify
- committer
- planner

How it Works



How it Works

Terminal

```
> mkdir flipr
> cd flipr
> git init .
Initialized empty Git repository in /flipr/.git/
> touch README.md
> git add .
> git commit -am 'First post!'

>
```

How it Works

Terminal

```
> mkdir flipr
> cd flipr
> git init .
Initialized empty Git repository in /flipr/.git/
> touch README.md
> git add .
> git commit -am 'First post!'

> sqitch --engine pg init flipr --uri https://github
Created sqitch.conf
Created sqitch.plan
Created deploy/
Created revert/
Created verify/
>
```

How it Works

Terminal

```
> mkdir flipr
> cd flipr
> git init .
Initialized empty Git repository in /flipr/.git/
> touch README.md
> git add .
> git commit -am 'First post!'

> sqitch --engine pg init flipr --uri https://github
Created sqitch.conf
Created sqitch.plan
Created deploy/
Created revert/
Created verify/
>
```

How it Works

Terminal

```
> mkdir flipr
> cd flipr
> git init .
Initialized empty Git repository in /flipr/.git/
> touch README.md
> git add .
> git commit -am 'First post!'

> sqitch --engine pg init flipr --uri https://github
Created sqitch.conf
Created sqitch.plan
Created deploy/
Created revert/
Created verify/
> emacs sqitch.conf
```

sqitch.conf

Emacs

```
[core]
    engine = pg
     # plan_file = sqitch.plan
     # top_dir = .
     # deploy_dir = deploy
     # revert_dir = revert
     # test_dir = test
     # extension = sql
# [core "pg"]
     # db_name =
     # client = /usr/local/pgsql/bin/psql
     # sqitch_schema = sqitch
     # password =
     # port =
     # host =
     # username =
```

--:-- sqitch.conf All (SQL[ansi])

sqitch.conf

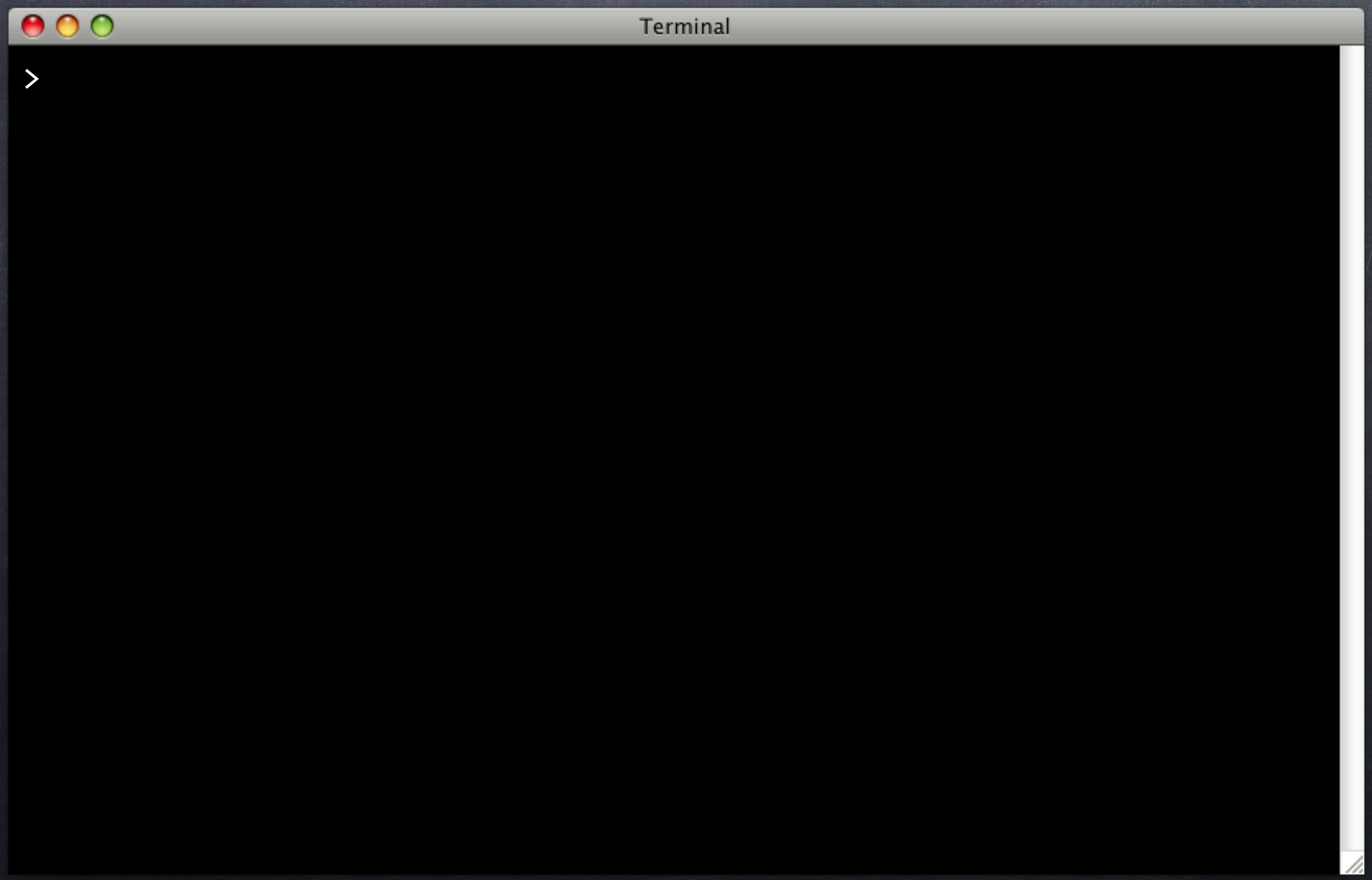
Emacs

```
[core]
    engine = pg
    # plan_file = s
    # top_dir = .
    # deploy_dir = deploy
    # revert_dir = revert
    # test_dir = test
    # extension = sql
# [core "pg"]
    # db_name =
    # client = /usr/local/pgsql/bin/psql
    # sqitch_schema = sqitch
    # password =
    # port =
    # host =
    # username =
```

--:-- sqitch.conf All (SQL[ansi])

--
engine pg

Add User Config



Add User Config

Terminal

```
> sqitch config --user core.pg.client /opt/local/pgsql/bin/psql
>
```

Add User Config

Terminal

```
> sqitch config --user core.pg.client /opt/local/pgsql/bin/psql
>
```

Add User Config

Terminal

```
> sqitch config --user core.pg.client /opt/local/pgsql/bin/psql
> sqitch config --user user.name 'Marge N. O'Vera'
>
```

Add User Config

Terminal

```
> sqitch config --user core.pg.client /opt/local/pgsql/bin/psql
> sqitch config --user user.name 'Marge N. O'Vera'
> sqitch config --user user.email 'marge@example.com'
>
```

Add User Config

Terminal

```
> sqitch config --user core.pg.client /opt/local/pgsql/bin/psql
> sqitch config --user user.name 'Marge N. O'Vera'
> sqitch config --user user.email 'marge@example.com'
> emacs ~/.sqitch/sqitch.conf
```

~/.sqitch/sqitch.conf

Emacs

```
[core "pg"]
    client = /opt/local/pgsql/bin/psql
[user]
    name = Marge N. O'Vera
    email = marge@example.com
```

--:-- ~/.sqitch/sqitc All (SQL[ansi])

~/.sqitch/sqitch.conf

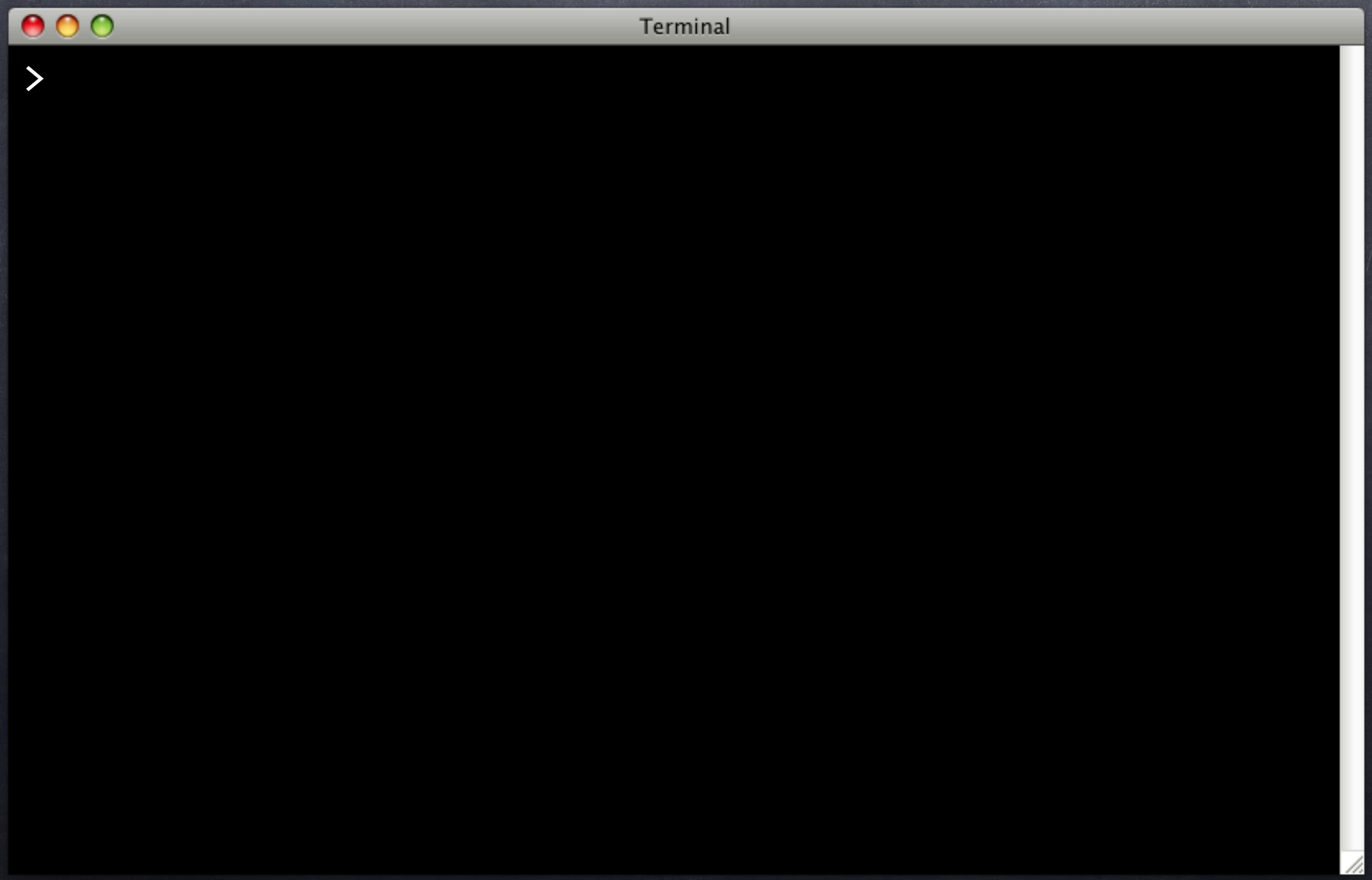
Emacs

```
[core "pg"]
    client = /opt/local/pgsql/bin/psql
[user]
    name = Marge N. O'Vera
    email = marge@example.com
```

Good for all projects

--:-- ~/.sqitch/sqitc All (SQL[ansi])

What's the Plan Man?



What's the Plan Man?

Terminal

```
> emacs sqitch.plan
```

sqitch.plan

Emacs

```
%syntax-version=1.0.0-b1
%project=flipr
%uri=https://github.com/theory/sqitch-intro/
```

--:-- sqitch.plan All (SQL[ansi])

sqitch.plan

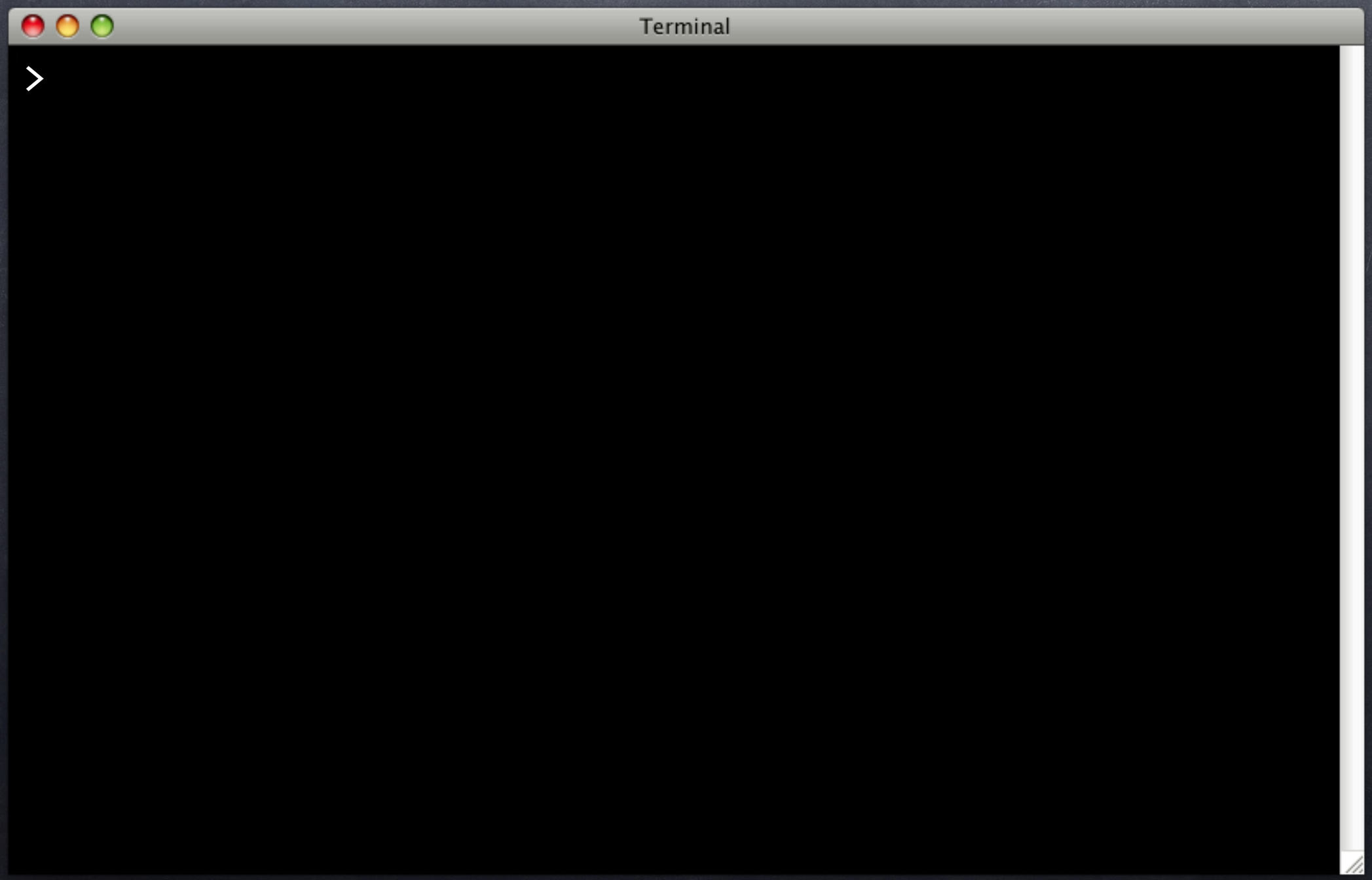
Emacs

```
%syntax-version=1.0.0-b1
%project=flipr
%uri=https://github.com/theory/sqitch-intro/
```

Identified

--:-- sqitch.plan All (SQL[ansi])

Make It So

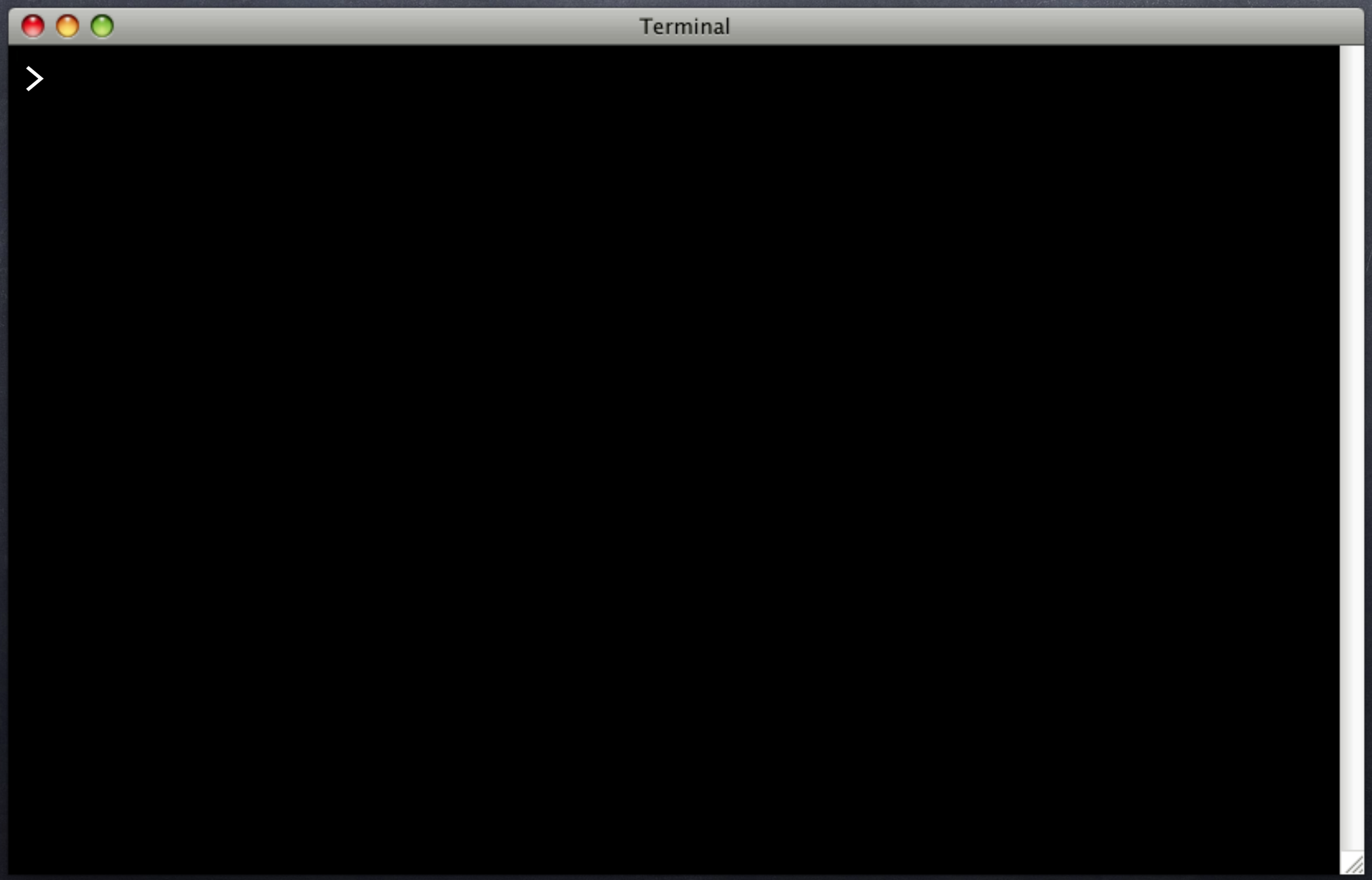


Make It So

Terminal

```
> git add .
> git commit -am 'Initialize Sqitch configuration.'
[master be9ab3e] Initialize Sqitch configuration.
 2 files changed, 23 insertions(+)
 create mode 100644 sqitch.conf
 create mode 100644 sqitch.plan
>
```

First Deployment



First Deployment

Terminal

```
> sqitch add appschema -n 'Adds flipr app schema.'
Created deploy/appschema.sql
Created revert/appschema.sql
Created verify/appschema.sql
Added "appschema" to sqitch.plan
>
```

First Deployment

Terminal

```
> sqitch add appschema -n 'Adds flipr app schema.'
Created deploy/appschema.sql
Created revert/appschema.sql
Created verify/appschema.sql
Added "appschema" to sqitch.plan
>
```

First Deployment

Terminal

```
> sqitch add appschema -n 'Adds flipr app schema.'
Created deploy/appschema.sql
Created revert/appschema.sql
Created verify/appschema.sql
Added "appschema" to sqitch.plan
>
```

First Deployment

Terminal

```
> sqitch add appschema -n 'Adds flipr app schema.'
Created deploy/appschema.sql
Created revert/appschema.sql
Created verify/appschema.sql
Added "appschema" to sqitch.plan
> emacs deploy/appschema.sql
```

deploy/appschema.sql

Emacs

```
-- Deploy appschema

BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

--:-- deploy/appschem All (SQL[ansi])

deploy/appschema.sql

Emacs

```
-- Deploy appschema

BEGIN;

CREATE SCHEMA flipr;

COMMIT;
```

--:-- deploy/appschem All (SQL[ansi])

First Deployment

Terminal

```
> sqitch add appschema -n 'Adds flipr app schema.'
Created deploy/appschema.sql
Created revert/appschema.sql
Created test/appschema.sql
Added "appschema" to sqitch.plan
> emacs deploy/appschema.sql
>
```

First Deployment

Terminal

```
> sqitch add appschema -n 'Adds flipr app schema.'
Created deploy/appschema.sql
Created revert/appschema.sql
Created test/appschema.sql
Added "appschema" to sqitch.plan
> emacs deploy/appschema.sql
> emacs revert/appschema.sql
```

revert/appschema.sql

Emacs

```
-- Revert appschema

BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

--:-- revert/appschem All (SQL[ansi])

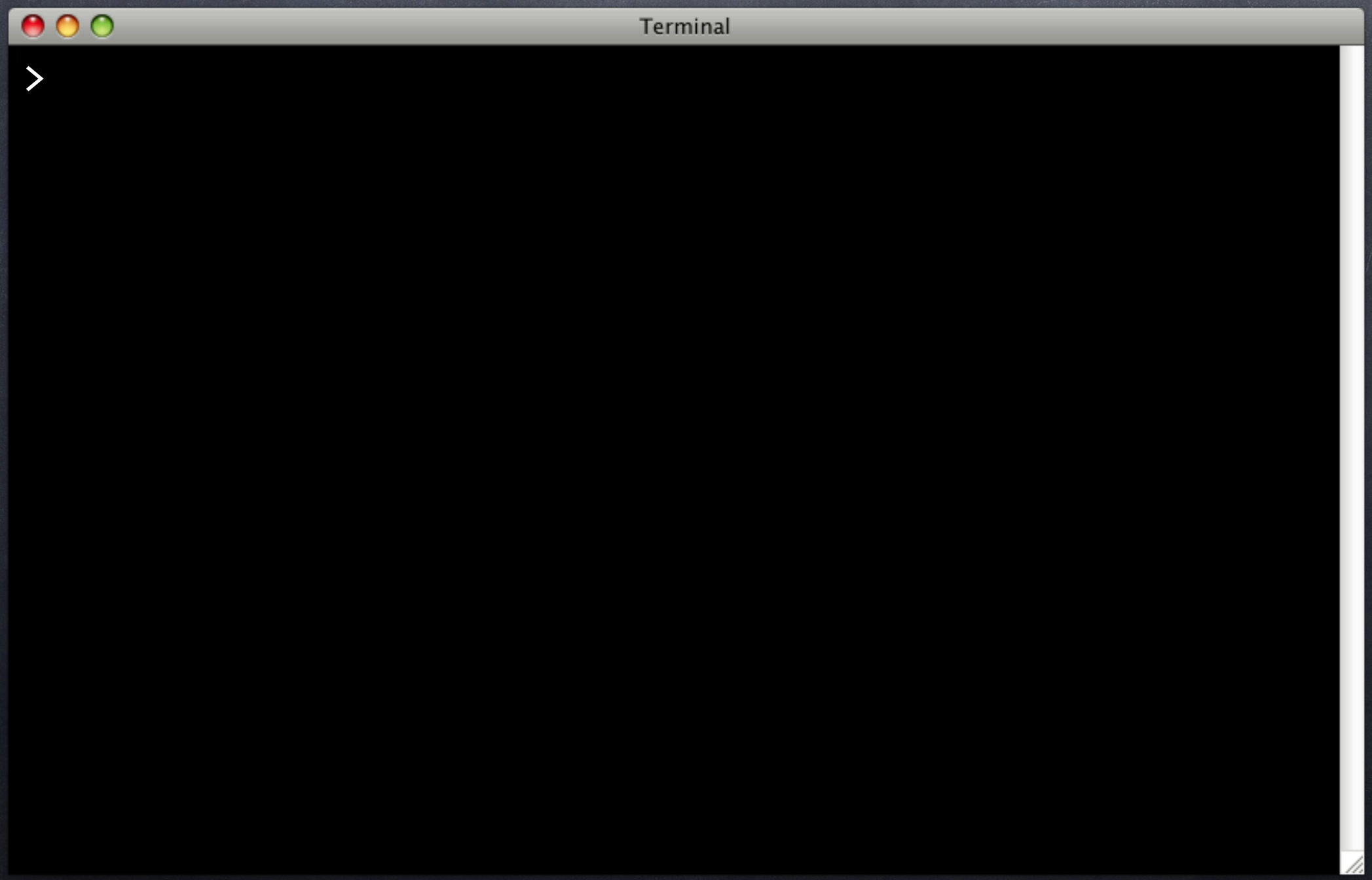
revert/appschema.sql

Emacs

```
-- Revert appschema

BEGIN;

DROP SCHEMA flipr;

COMMIT;
```

--:-- revert/appschem All (SQL[ansi])

Make it So!



Make it So!

Terminal

```
> createdb flipr_test
> sqitch --db-name flipr_test deploy
Adding metadata tables to flipr_test
Deploying to flipr_test
  + appschema .. ok
>
```

Make it So!

Terminal

```
> createdb flipr_test
> sqitch --db-name flipr_test deploy
Adding metadata tables to flipr_test
Deploying to flipr_test
  + appschema .. ok
>
```

Make it So!

Terminal

```
> createdb flipr_test
> sqitch --db-name flipr_test deploy
Adding metadata tables to flipr_test
Deploying to flipr_test
  + appschema .. ok
>
```

Make it So!

Terminal

```
> createdb flipr_test
> sqitch --db-name flipr_test deploy
Adding metadata tables to flipr_test
Deploying to flipr_test
  + appschema .. ok
>
```

Make it So!

Terminal

```
> createdb flipr_test
> sqitch --db-name flipr_test deploy
Adding metadata tables to flipr_test
Deploying to flipr_test
  + appschema .. ok
> psql -d flipr_test -c '\dn flipr'
List of schemas
 Name  | Owner
-------+-------
 flipr | marge
>
```

Make it So!

Terminal

```
> createdb flipr_test
> sqitch --db-name flipr_test deploy
Adding metadata tables to flipr_test
Deploying to flipr_test
  + appschema .. ok
> psql -d flipr_test -c '\dn flipr'
List of schemas
 Name  | Owner
-------+-------
 flipr | marge
>
```

Trust, but verify

Make it So!

Terminal

```
> createdb flipr_test
> sqitch --db-name flipr_test deploy
Adding metadata tables to flipr_test
Deploying to flipr_test
  + appschema .. ok
> psql -d flipr_test -c '\dn flipr'
List of schemas
 Name  | Owner
-------+-------
 flipr | marge
> emacs verify/appschema.sql
```

verify/appschema.sql

Emacs

```
-- Verify appschema

BEGIN;

-- XXX Add verifications here.

COMMIT;
```

--:-- verify/appschem All (SQL[ansi])

verify/appschema.sql

Emacs

```
-- Verify appschema

BEGIN;

SELECT pg_catalog.has_schema_privilege('nada', 'usage');

COMMIT;
```

--:-- verify/appschem All (SQL[ansi])

verify/appschema.sql

Emacs

```sql
-- Verify appschema

BEGIN;

SELECT pg_catalog.has_schema_privilege('nada', 'usage');

COMMIT;
```

Let's try it, first

--:-- verify/appschem All (SQL[ansi])

verify/appschema.sql

Emacs

```
-- Verify appschema

BEGIN;

SELECT pg_catalog.has_schema_privilege('nada', 'usage');

COMMIT;
```

--:-- verify/appschem All (SQL[ansi])

Trust, But Verify

Terminal

```
> emacs verify/appschema.sql
>
```

Trust, But Verify

Terminal

```
> emacs verify/appschema.sql
> sqitch --db-name flipr_test verify
Verifying flipr_test
  * appschema .. psql:verify/appschema.sql:5:
ERROR:  schema "nada" does not exist
# Verify script "verify/appschema.sql" failed.
not ok

Verify Summary Report
---------------------
Changes: 1
Errors:  1
Verify failed
>
```

Trust, But Verify

Terminal

```
> emacs verify/appschema.sql
> sqitch --db-name flipr_test verify
Verifying flipr_test
  * appschema .. psql:verify/appschema.sql:5:
ERROR:  schema "nada" does not exist
# Verify script "verify/appschema.sql" failed.
not ok

Verify Summary Report
---------------------
Changes: 1
Errors:  1
Verify failed
>
```

Trust, But Verify

Terminal

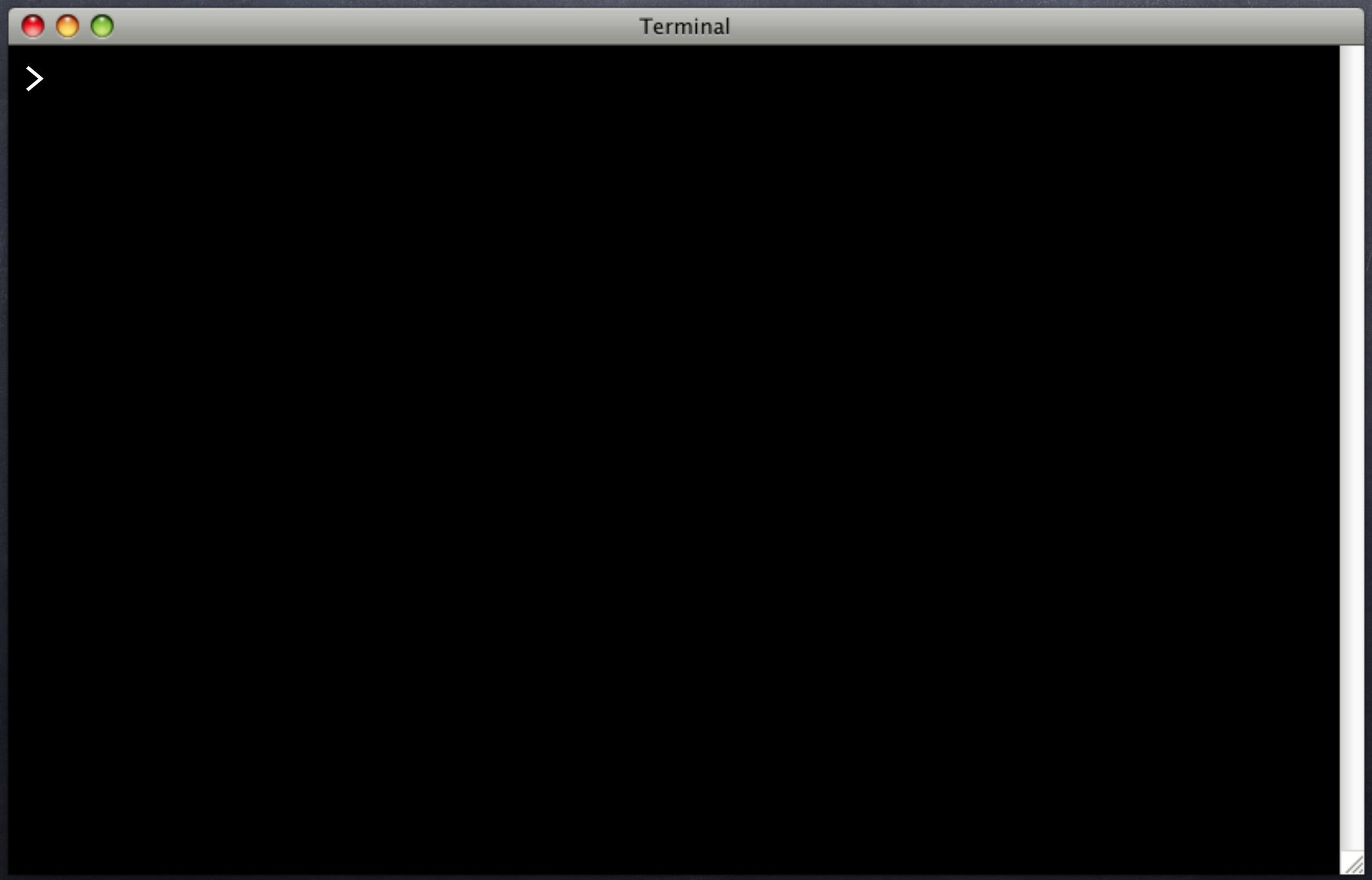
```
> emacs verify/appschema.sql
> sqitch --db-name flipr_test verify
Verifying flipr_test
  * appschema .. psql:verify/appschema.sql:5:
ERROR:  schema "nada" does not exist
# Verify script "verify/appschema.sql" failed.
not ok

Verify Summary Report
---------------------
Changes: 1
Errors:  1
Verify failed
>
```

Trust, But Verify

Terminal

```
> emacs verify/appschema.sql
> sqitch --db-name flipr_test verify
Verifying flipr_test
  * appschema .. psql:verify/appschema.sql:5:
ERROR:  schema "nada" does not exist
# Verify script "verify/appschema.sql" failed.
not ok

Verify Summary Report
---------------------
Changes: 1
Errors:  1
Verify failed
>
```

Trust, But Verify



Trust, But Verify

Terminal

```
> perl -i -pe 's/nada/flipr/'
>
```

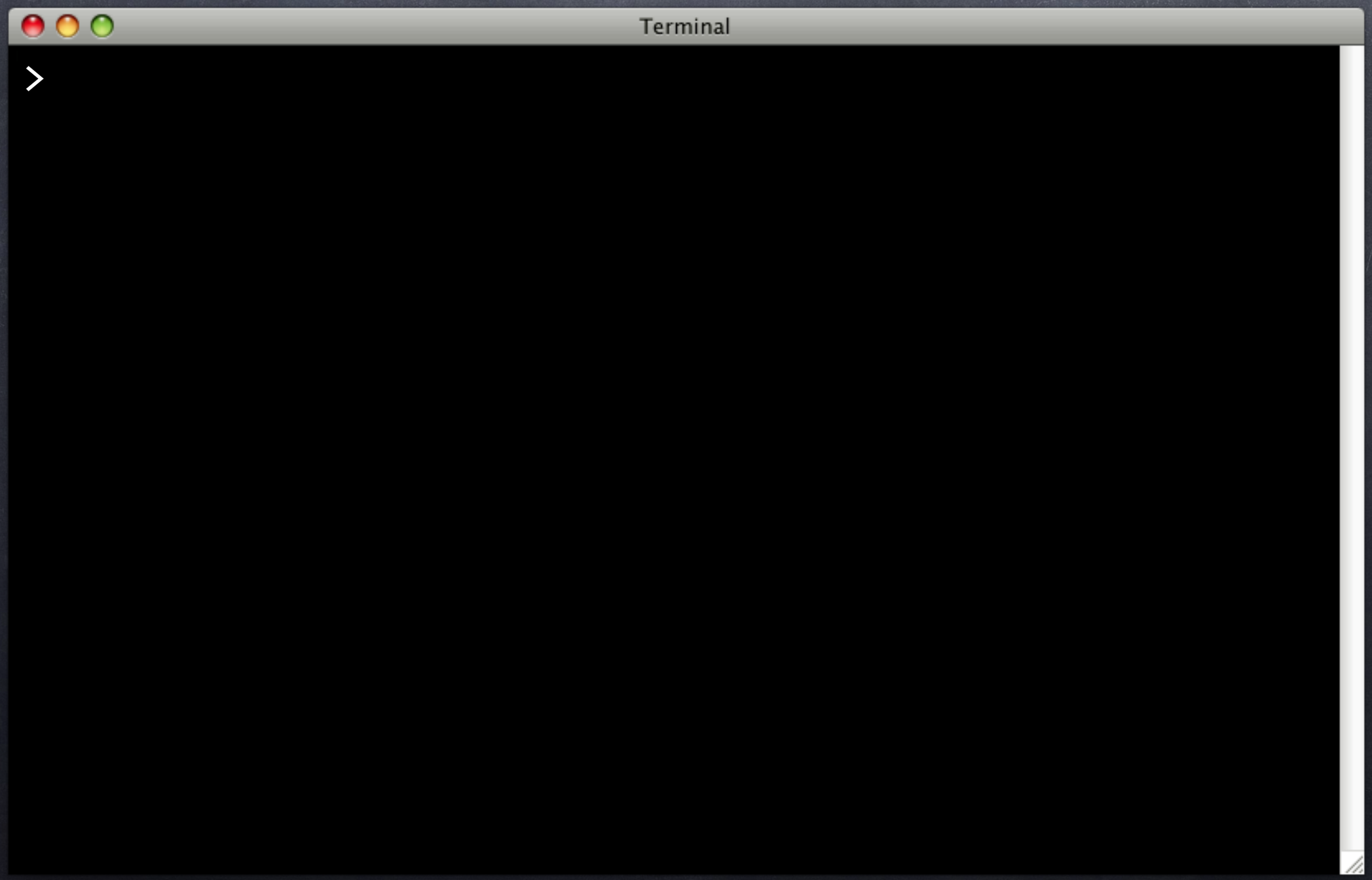
Trust, But Verify

Terminal

```
> perl -i -pe 's/nada/flipr/'
> sqitch --db-name flipr_test verify
Verifying flipr_test
  * appschema .. ok
Verify successful
>
```

Mo betta.

How's it Look?



How's it Look?

Terminal

```
> sqitch -d flipr_test status
# On database flipr_test
# Project:  flipr
# Change:   ae1263b92b3355cf705c7832271730fe4ccb3c77
# Name:     appschema
# Deployed: 2012-12-31 11:00:29 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

How's it Look?

Terminal

```
> sqitch -d flipr_test status
# On database flipr_test
# Project:  flipr
# Change:   ae1263b92b3355cf705c7832271730fe4ccb3c77
# Name:     appschema
# Deployed: 2012-12-31 11:00:29 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

How's it Look?

Terminal

```
> sqitch -d flipr_test status
# On database flipr_test
# Project:  flipr
# Change:   ae1263b92b3355cf705c78322271730fe4ccb3c77
# Name:     appschema
# Deployed: 2012-12-31 11:00:29 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

How's it Look?

Terminal

```
> sqitch -d flipr_test status
# On database flipr_test
# Project:  flipr
# Change:   ae1263b92b3355cf705c7832271730fe4ccb3c77
# Name:     appschema
# Deployed: 2012-12-31 11:00:29 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

How's it Look?

Terminal

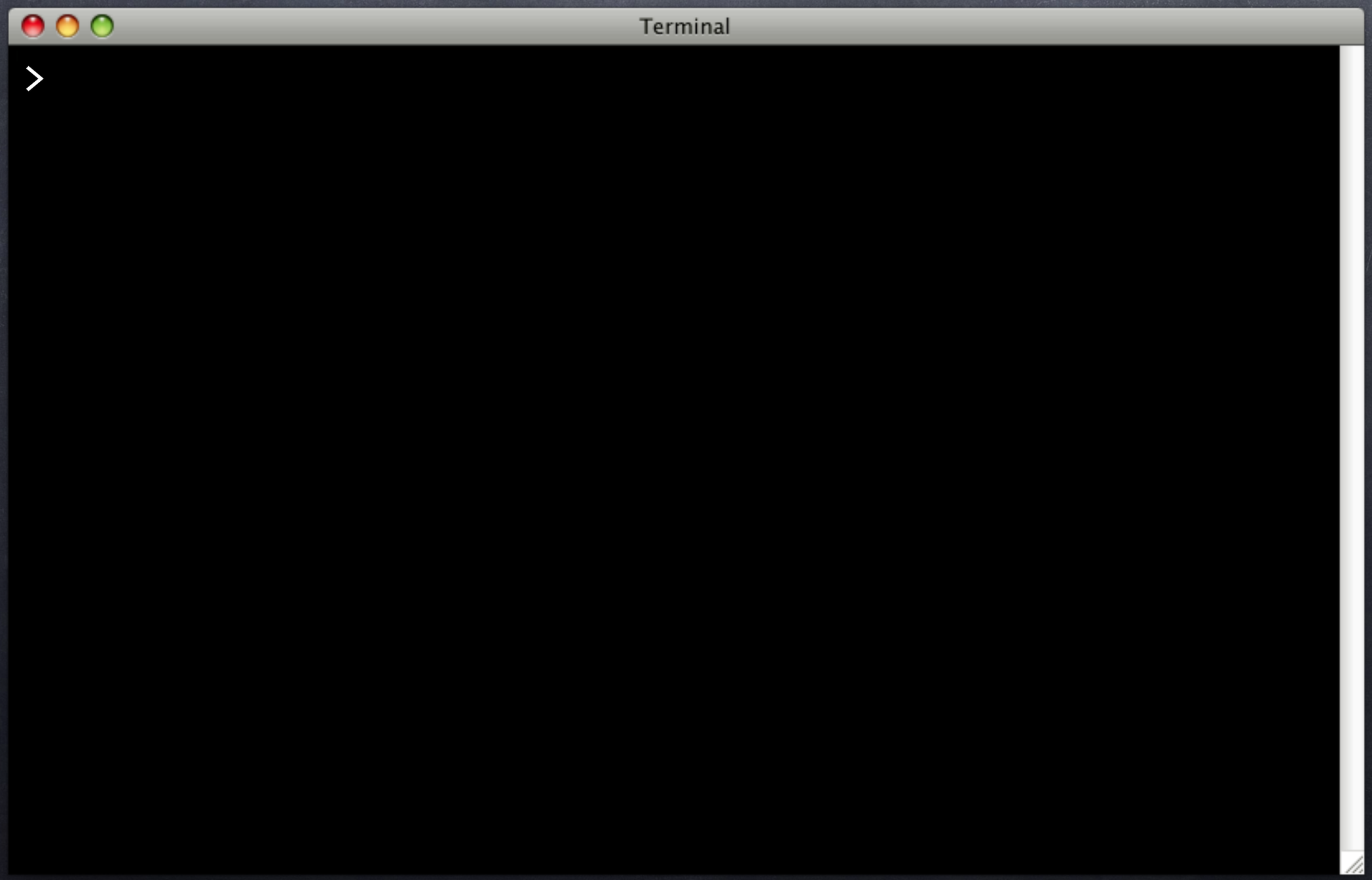
```
> sqitch -d flipr_test status
# On database flipr_test
# Project:  flipr
# Change:   ae1263b92b3355cf705c7832271730fe4ccb3c77
# Name:     appschema
# Deployed: 2012-12-31 11:00:29 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

How's it Look?

Terminal

```
> sqitch -d flipr_test status
# On database flipr_test
# Project:  flipr
# Change:   ae1263b92b3355cf705c7832271730fe4ccb3c77
# Name:     appschema
# Deployed: 2012-12-31 11:00:29 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

Go Back



Go Back

Terminal

```
> sqitch --db-name flipr_test revert
  Revert all changes from flipr_test? [Yes]
```

Go Back

Terminal

```
> sqitch --db-name flipr_test revert
  Revert all changes from flipr_test? [Yes]
```

Go Back

Terminal

```
> sqitch --db-name flipr_test revert
  Revert all changes from flipr_test? [Yes]
  - appschema .. ok
>
```

Go Back

Terminal

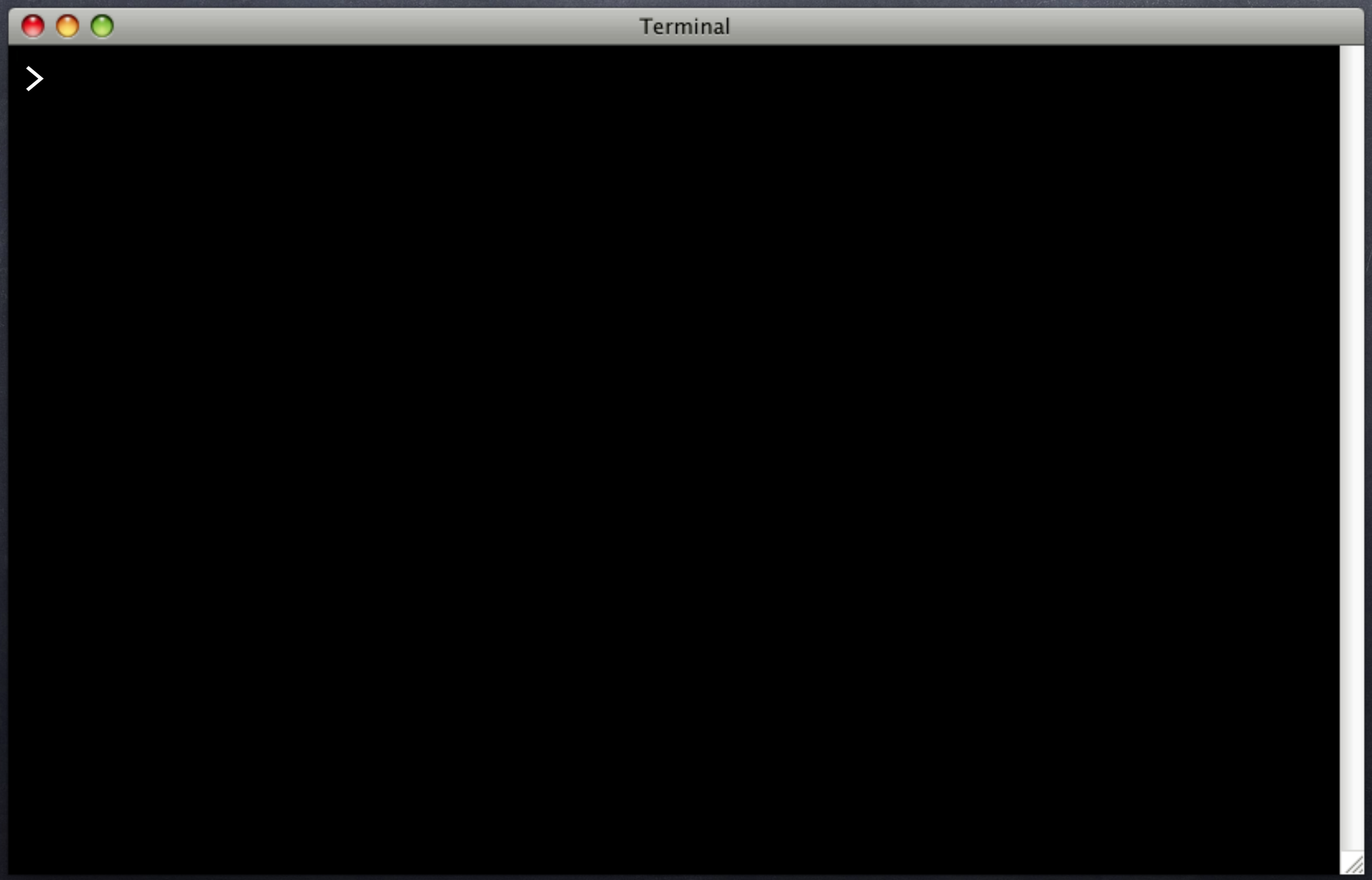
```
> sqitch --db-name flipr_test revert
  Revert all changes from flipr_test? [Yes]
  - appschema .. ok
> psql -d flipr_test -c '\dn flipr'
             List of roles
List of schemas
 Name | Owner
------+-------
>
```

Go Back

Terminal

```
> sqitch --db-name flipr_test revert
  Revert all changes from flipr_test? [Yes]
  - appschema .. ok
> psql -d flipr_test -c '\dn flipr'
             List of roles
List of schemas
 Name | Owner
------+-------
> sqitch -d flipr_test status
# On database flipr_test
No changes deployed
>
```

History



History

Terminal

```
> sqitch -d flipr_test log
On database flipr_test
Revert ae1263b92b3355cf705c7832271730fe4ccb3c77
Name:       appschema
Committer: Marge N. O'Vera <marge@example.com>
Date:       2012-12-31 11:01:22 -0800

    App schema for all flipr objects.

Deploy ae1263b92b3355cf705c7832271730fe4ccb3c77
Name:       appschema
Committer: Marge N. O'Vera <marge@example.com>
Date:       2012-12-31 11:00:29 -0800

    App schema for all flipr objects.
```

History

Terminal

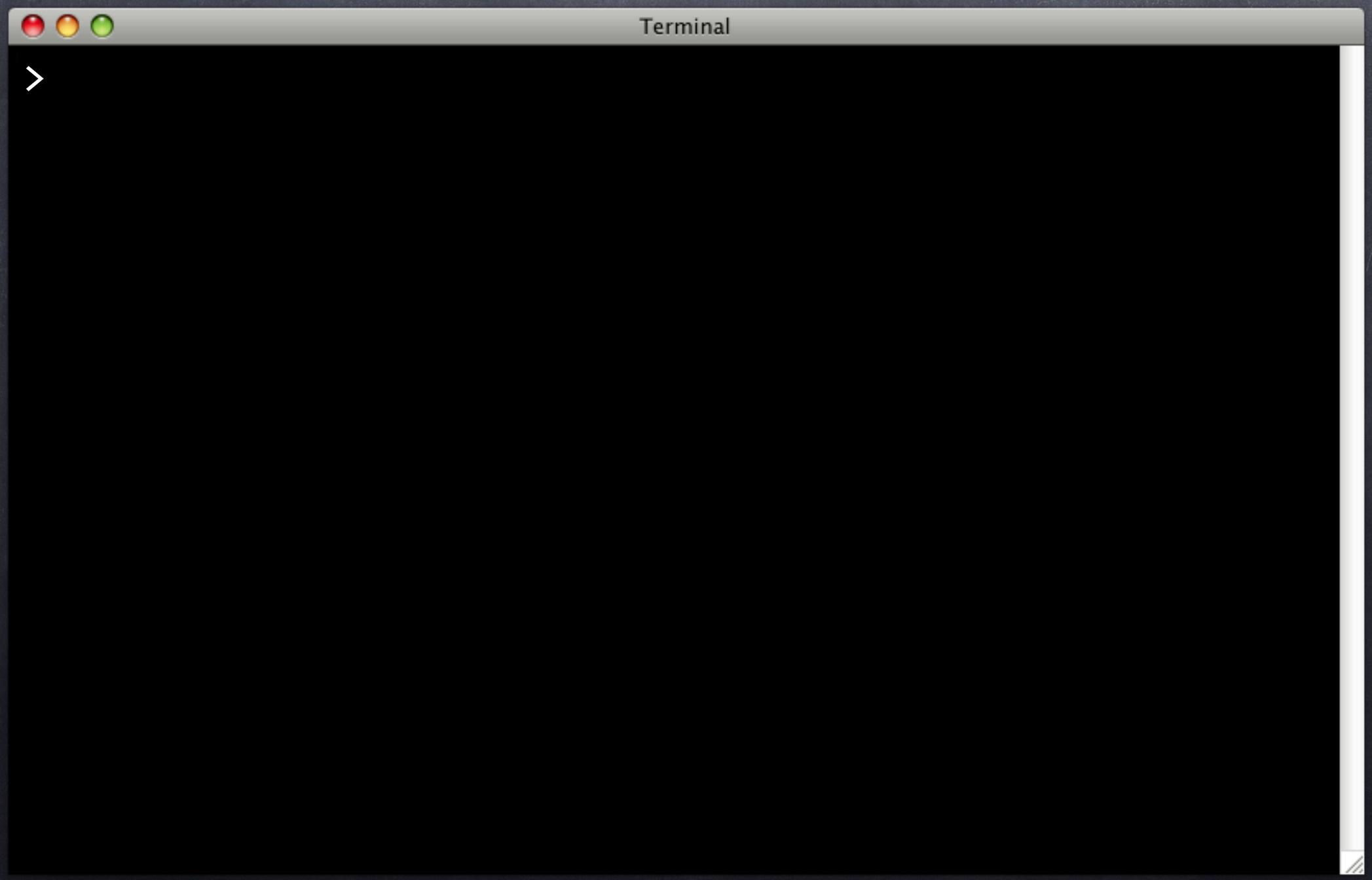
```
> sqitch -d flipr_test log
On database flipr_test
Revert ae1263b92b3355cf705c7832271730fe4ccb3c77
Name:       appschema
Committer: Marge N. O'Vera <marge@example.com>
Date:       2012-12-31 11:01:22 -0800

    App schema for all flipr objects.

Deploy ae1263b92b3355cf705c7832271730fe4ccb3c77
Name:       appschema
Committer: Marge N. O'Vera <marge@example.com>
Date:       2012-12-31 11:00:29 -0800

    App schema for all flipr objects.
```

History

Terminal

```
> sqitch -d flipr_test log
On database flipr_test
Revert ae1263b92b3355cf705c7832271730fe4ccb3c77
Name:       appschema
Committer: Marge N. O'Vera <marge@example.com>
Date:       2012-12-31 11:01:22 -0800

    App schema for all flipr objects.

Deploy ae1263b92b3355cf705c7832271730fe4ccb3c77
Name:       appschema
Committer: Marge N. O'Vera <marge@example.com>
Date:       2012-12-31 11:00:29 -0800

    App schema for all flipr objects.
```

Commit It!



Commit It!

Terminal

```
> git add .
> git commit -m 'Add flipr schema.'
[master e6464c0] Add flipr schema.
 4 files changed, 22 insertions(+)
 create mode 100644 deploy/appschema.sql
 create mode 100644 revert/appschema.sql
 create mode 100644 verify/appschema.sql
>
```

Commit It!

Terminal

```
> git add .
> git commit -m 'Add flipr schema.'
[master e6464c0] Add flipr schema.
 4 files changed, 22 insertions(+)
 create mode 100644 deploy/appschema.sql
 create mode 100644 revert/appschema.sql
 create mode 100644 verify/appschema.sql
> sqitch --db-name flipr_test deploy --verify
Deploying changes to flipr_test
  + appschema .. ok
>
```

Commit It!

Terminal

```
> git add .
> git commit -m 'Add flipr schema.'
[master e6464c0] Add flipr schema.
 4 files changed, 22 insertions(+)
 create mode 100644 deploy/appschema.sql
 create mode 100644 revert/appschema.sql
 create mode 100644 verify/appschema.sql
> sqitch --db-name flipr_test deploy --verify
Deploying changes to flipr_test
  + appschema .. ok
>
```

Integrated!

Commit It!

Terminal

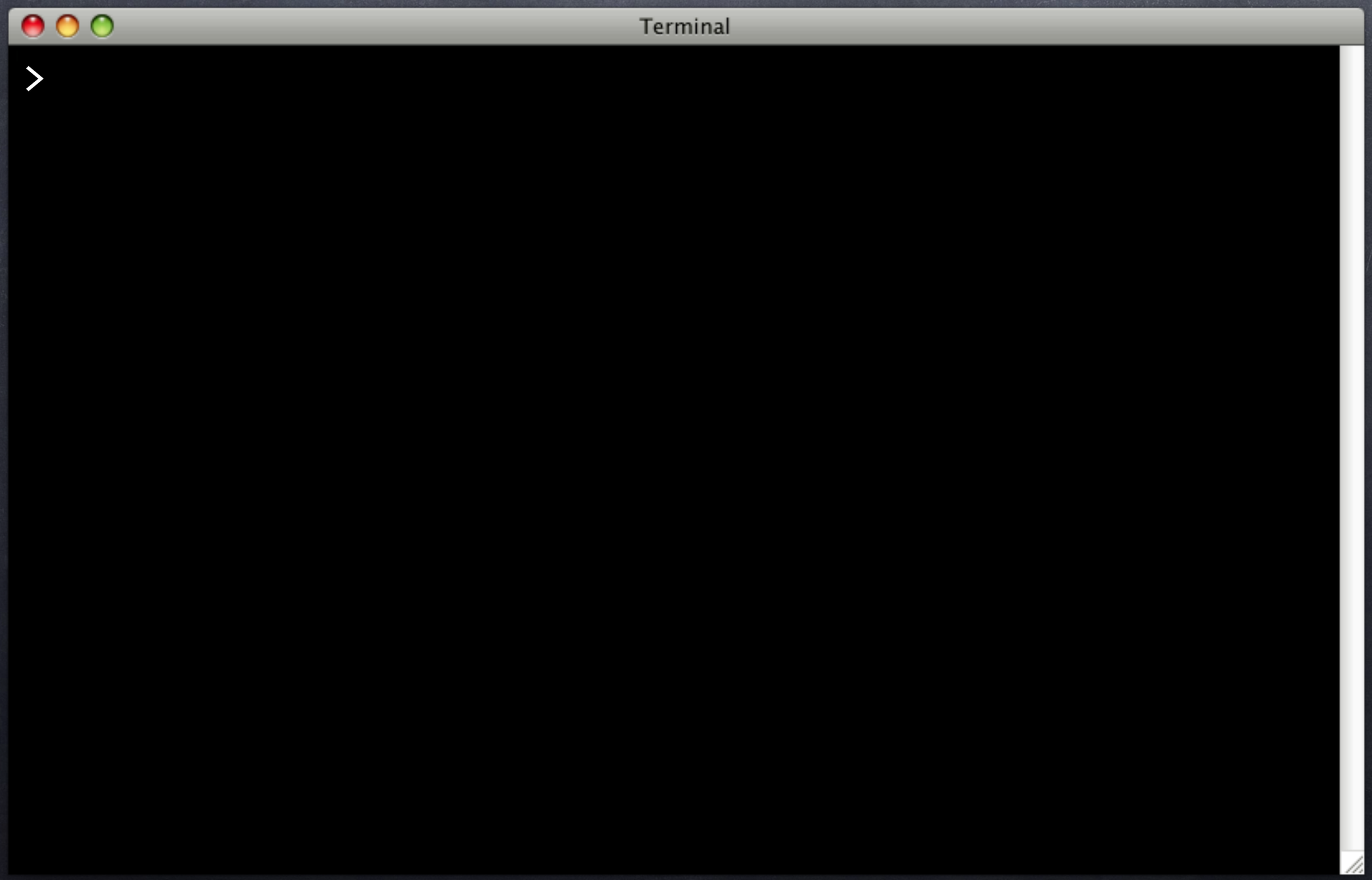
```
> git add .
> git commit -m 'Add flipr schema.'
[master e6464c0] Add flipr schema.
 4 files changed, 22 insertions(+)
 create mode 100644 deploy/appschema.sql
 create mode 100644 revert/appschema.sql
 create mode 100644 verify/appschema.sql
> sqitch --db-name flipr_test deploy --verify
Deploying changes to flipr_test
  + appschema .. ok
>
```

Commit It!

Terminal

```
> git add .
> git commit -m 'Add flipr schema.'
[master e6464c0] Add flipr schema.
 4 files changed, 22 insertions(+)
 create mode 100644 deploy/appschema.sql
 create mode 100644 revert/appschema.sql
 create mode 100644 verify/appschema.sql
> sqitch --db-name flipr_test deploy --verify
Deploying changes to flipr_test
  + appschema .. ok
> psql -d flipr_test -c '\dn flipr'
List of schemas
 Name  | Owner
-------+-------
 flipr | marge
```

Status Update

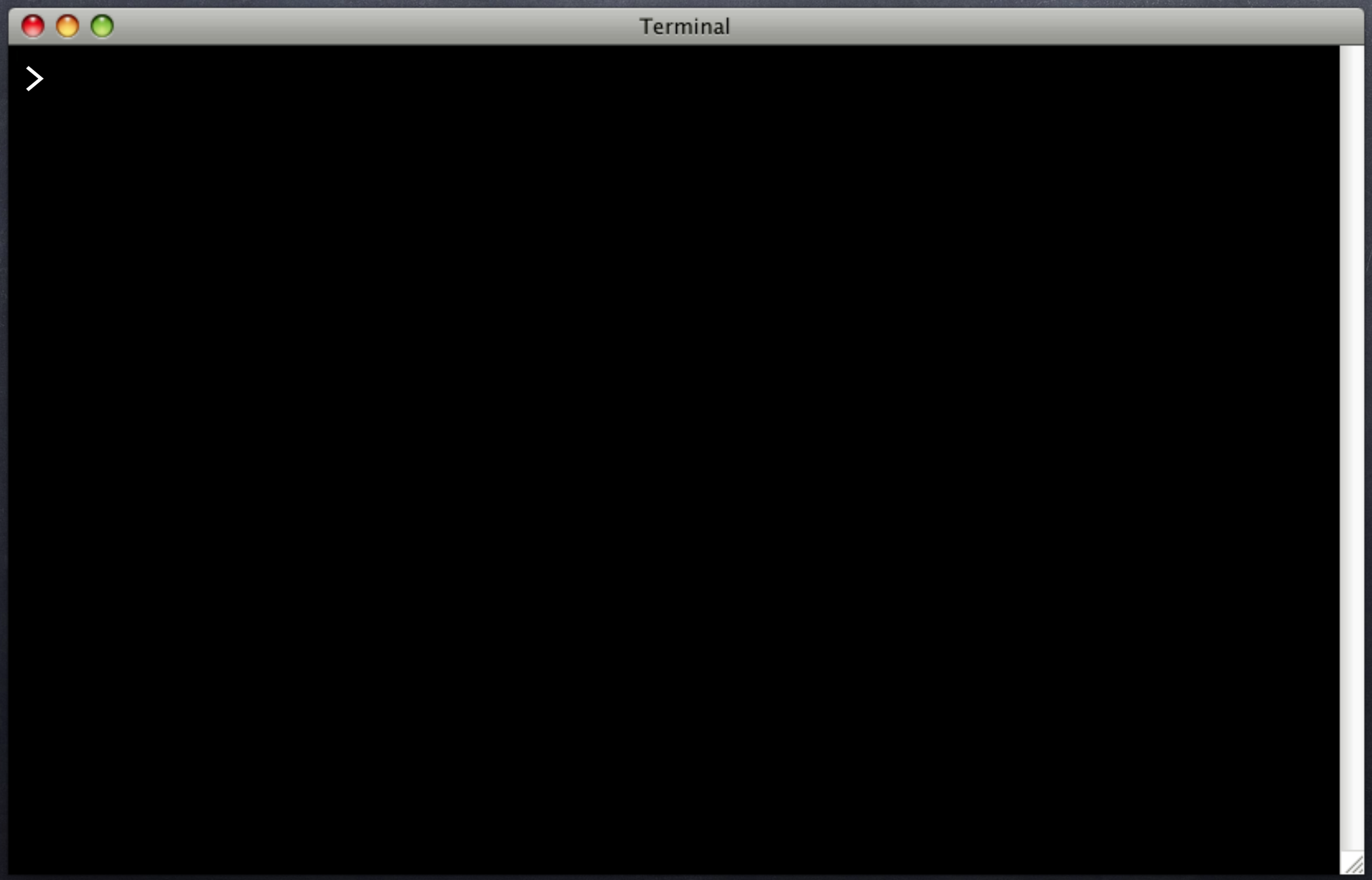


Status Update

Terminal

```
> sqitch -d flipr_test status
# On database flipr_test
# Project:  flipr
# Change:   ae1263b92b3355cf705c7832271730fe4ccb3c77
# Name:     appschema
# Deployed: 2012-12-31 11:03:18 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

Save My Fingers



Save My Fingers

Terminal

```
> sqitch config core.pg.db_name flipr_test
>
```

Save My Fingers

Terminal

```
> sqitch config core.pg.db_name flipr_test
> sqitch status
# On database flipr_test
# Project:  flipr
# Change:   ae1263b92b3355cf705c                          3c77
# Name:     appschema
# Deployed: 2012-12-31 11:03:18 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

No --db-name

Save My Fingers

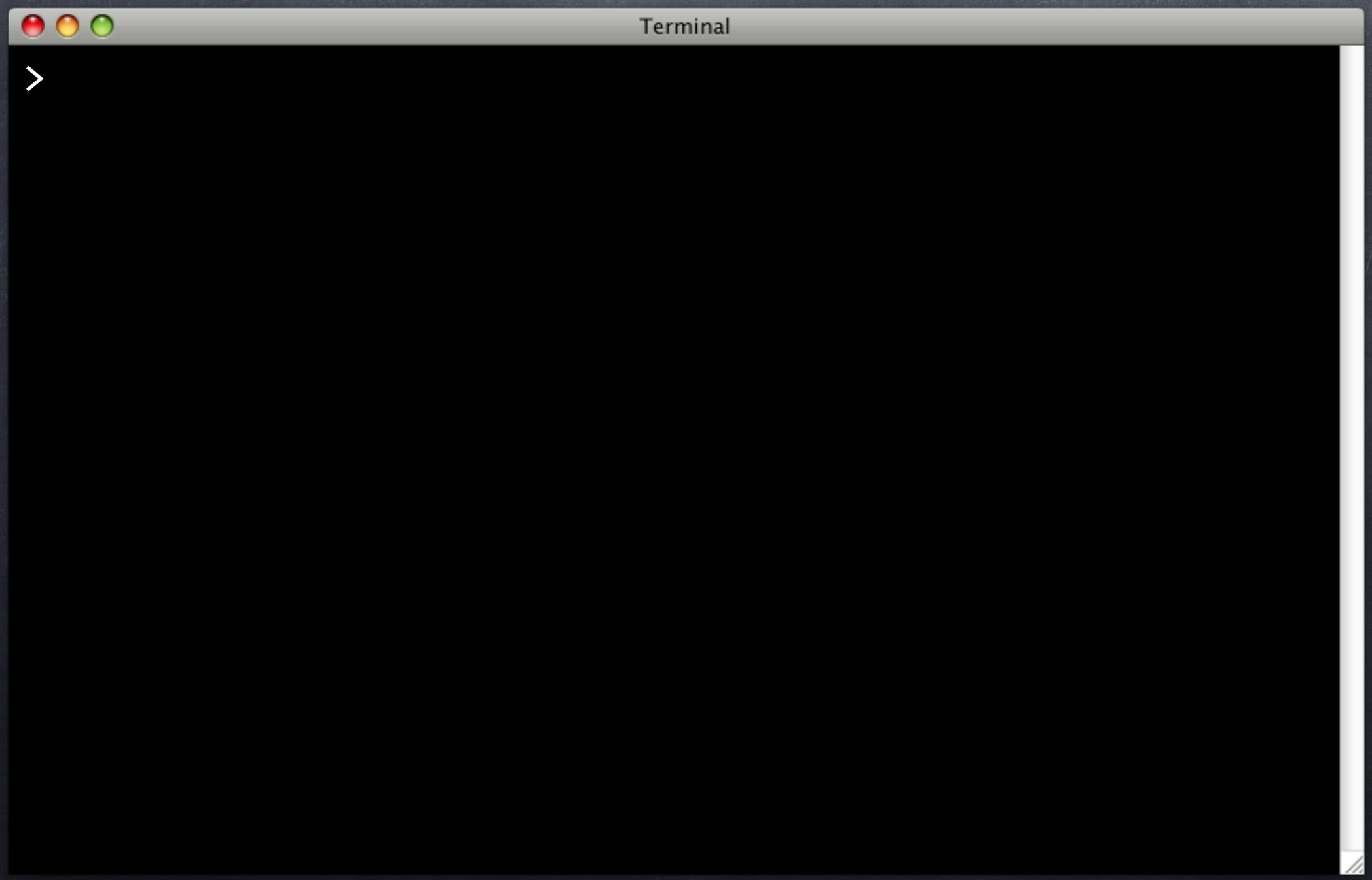
Terminal

```
> sqitch config core.pg.db_name flipr_test
> sqitch status
# On database flipr_test
# Project:  flipr
# Change:   ae1263b92b3355cf705c                    3c77
# Name:     appschema
# Deployed: 2012-12-31 11:03:18 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
> sqitch config --bool deploy.verify true
> sqitch config --bool rebase.verify true
>
```

No --db-name

Always verify.

Dependencies!



Dependencies!

Terminal

```
> sqitch add users --requires appschema \
  -n 'Creates table to track our users.'
Created deploy/users.sql
Created revert/users.sql
Created verify/users.sql
Added "users [appschema]" to sqitch.plan
>
```

Dependencies!

Terminal

```
> sqitch add users --requires appschema \
  -n 'Creates table to track our users.'
Created deploy/users.sql
Created revert/users.sql
Created verify/users.sql
Added "users [appschema]" to sqitch.plan
>
```

Dependencies!

Terminal

```
> sqitch add users --requires appschema \
  -n 'Creates table to track our users.'
Created deploy/users.sql
Created revert/users.sql
Created verify/users.sql
Added "users [appschema]" to sqitch.plan
> emacs deploy/users.sql
```

deploy/users.sql

Emacs

```
-- Deploy users
-- requires: appschema

BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

--:-- deploy/users.sq All (SQL[ansi])

deploy/users.sql

Emacs

```
-- Deploy users
-- requires: appschema

BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

--:-- deploy/users.sq All (SQL[ansi])

deploy/users.sql

Emacs

```
-- Deploy users
-- requires: appschema

BEGIN;

CREATE TABLE flipr.users (
    nickname  TEXT        PRIMARY KEY,
    password  TEXT        NOT NULL,
    timestamp TIMESTAMPTZ NOT NULL DEFAULT NOW()
);

COMMIT;
```

--:-- deploy/users.sq All (SQL[ansi])

Dependencies!

Terminal

```
> sqitch add users --requires appschema \
  -n 'Creates table to track our users.'
Created deploy/users.sql
Created revert/users.sql
Created verify/users.sql
Added "users [appschema]" to sqitch.plan
> emacs deploy/users.sql
>
```

Dependencies!

Terminal

```
> sqitch add users --requires appschema \
  -n 'Creates table to track our users.'
Created deploy/users.sql
Created revert/users.sql
Created verify/users.sql
Added "users [appschema]" to sqitch.plan
> emacs deploy/users.sql
> emacs verify/users.sql
```

verify/users.sql

Emacs

```
-- Verify users

BEGIN;

-- XXX Add verifications here.

COMMIT;
```

--:-- verify/users.sq All (SQL[ansi])

verify/users.sql

Emacs

```
-- Verify users

BEGIN;

SELECT nickname, password, timestamp
  FROM flipr.users
 WHERE FALSE;

COMMIT;
```

--:-- verify/users.sq All (SQL[ansi])

Dependencies!

Terminal

```
> sqitch add users --requires appschema \
  -n 'Creates table to track our users.'
Created deploy/users.sql
Created revert/users.sql
Created verify/users.sql
Added "users [appschema]" to sqitch.plan
> emacs deploy/users.sql
> emacs verify/users.sql
>
```

Dependencies!

Terminal

```
> sqitch add users --requires appschema \
  -n 'Creates table to track our users.'
Created deploy/users.sql
Created revert/users.sql
Created verify/users.sql
Added "users [appschema]" to sqitch.plan
> emacs deploy/users.sql
> emacs verify/users.sql
> emacs revert/users.sql
```

revert/users.sql

Emacs

```
-- Revert users

BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

--:-- revert/users.sq All (SQL[ansi])

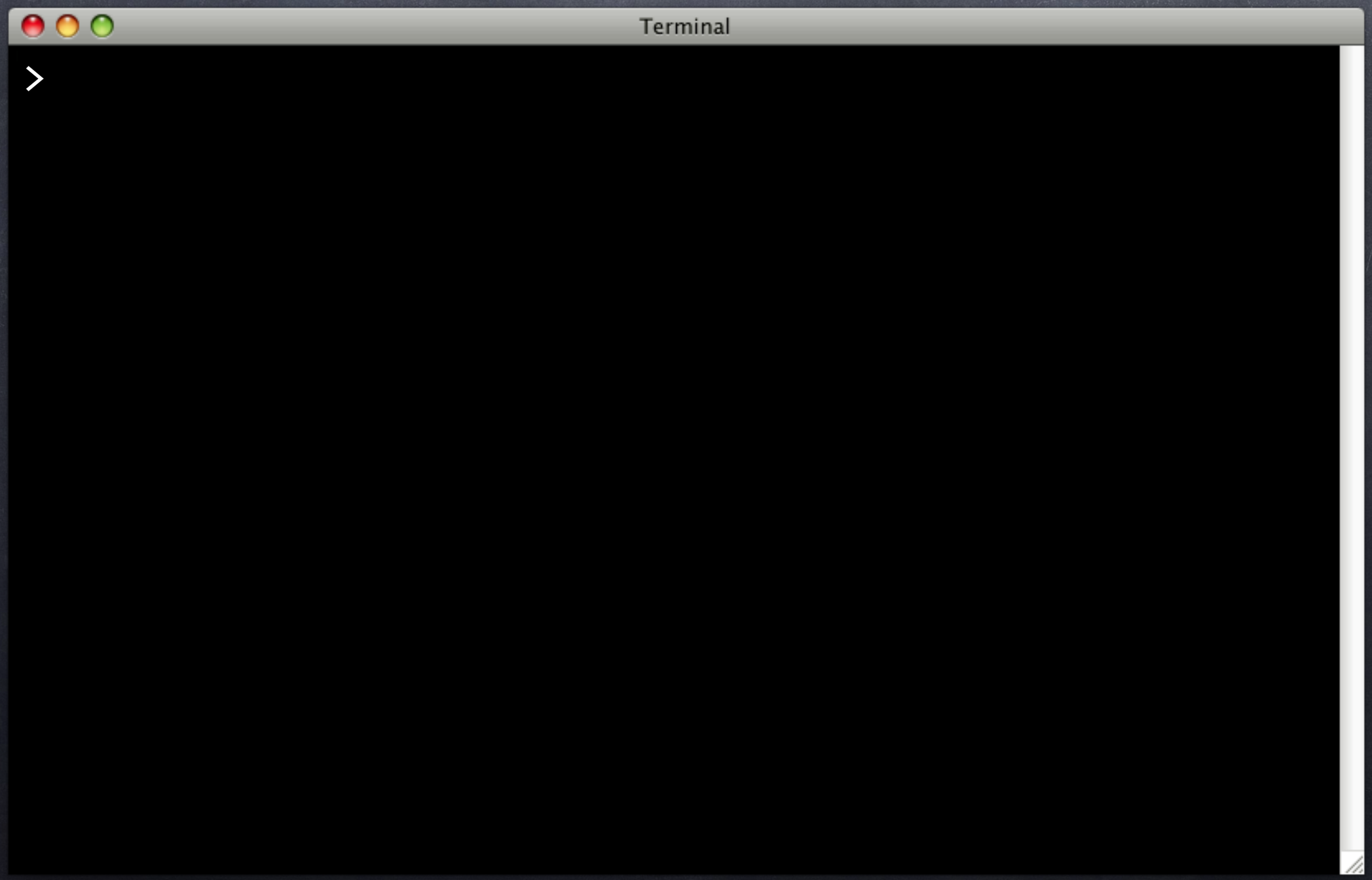
revert/users.sql

Emacs

```
-- Revert users

BEGIN;

DROP TABLE flipr.users;

COMMIT;
```

--:-- revert/users.sq All (SQL[ansi])

Make Users



Make Users

Terminal

```
> sqitch deploy
Deploying changes to flipr_test
  + users .. ok
>
```

Make Users

Terminal

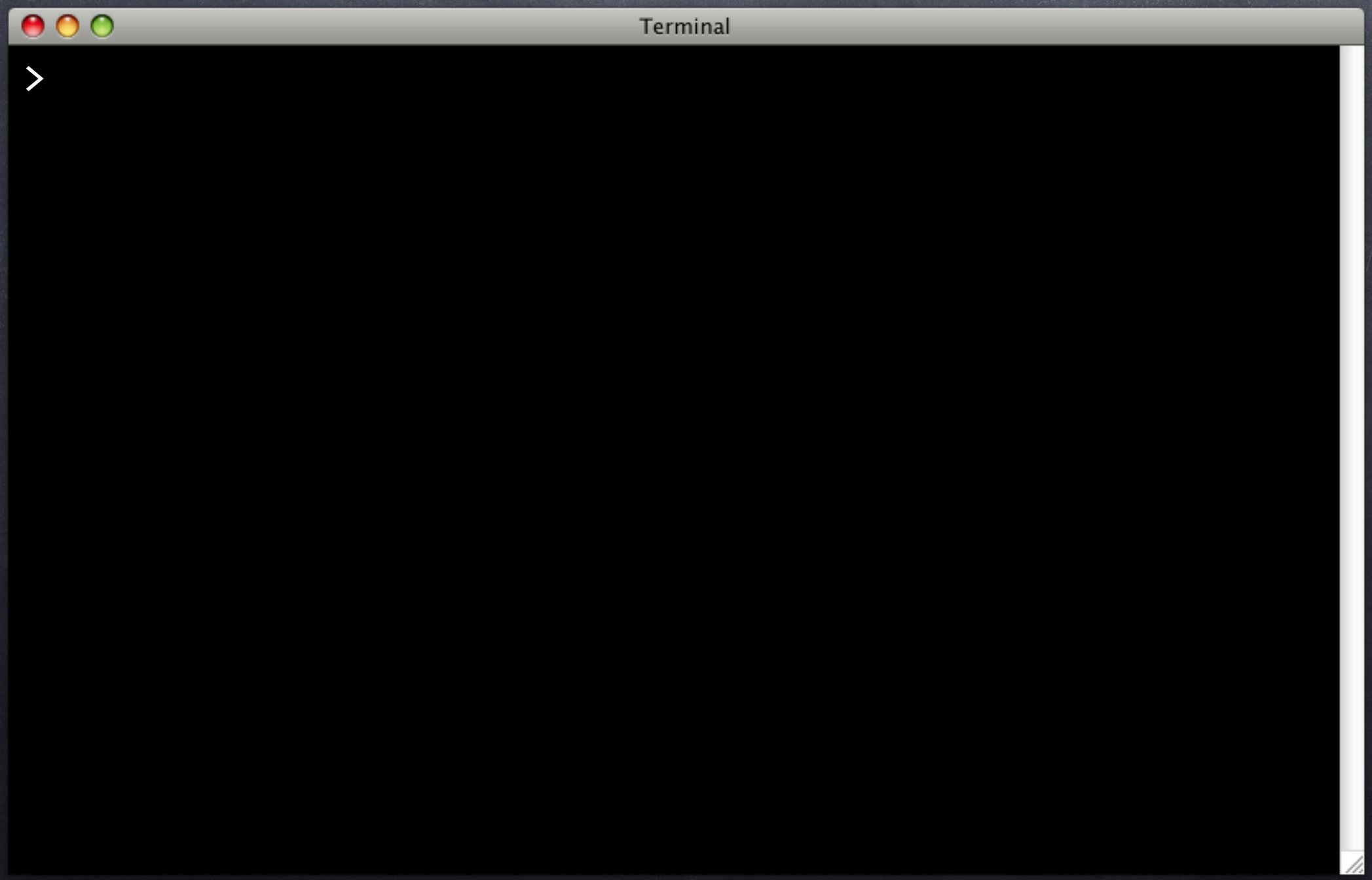
```
> sqitch deploy
Deploying changes to flipr_test
  + users .. ok
>
```

Make Users

Terminal

```
> sqitch deploy
Deploying changes to flipr_test
  + users .. ok
> psql -d flipr_test -c '\d flipr.users'
                    Table "flipr.users"
  Column   |      Type     |        Modifiers
-----------+---------------+-------------------------
 nickname  | text          | not null
 password  | text          | not null
 timestamp | timestamptz   | not null default now()
Indexes:
    "users_pkey" PRIMARY KEY, btree (nickname)
>
```

Verify Users

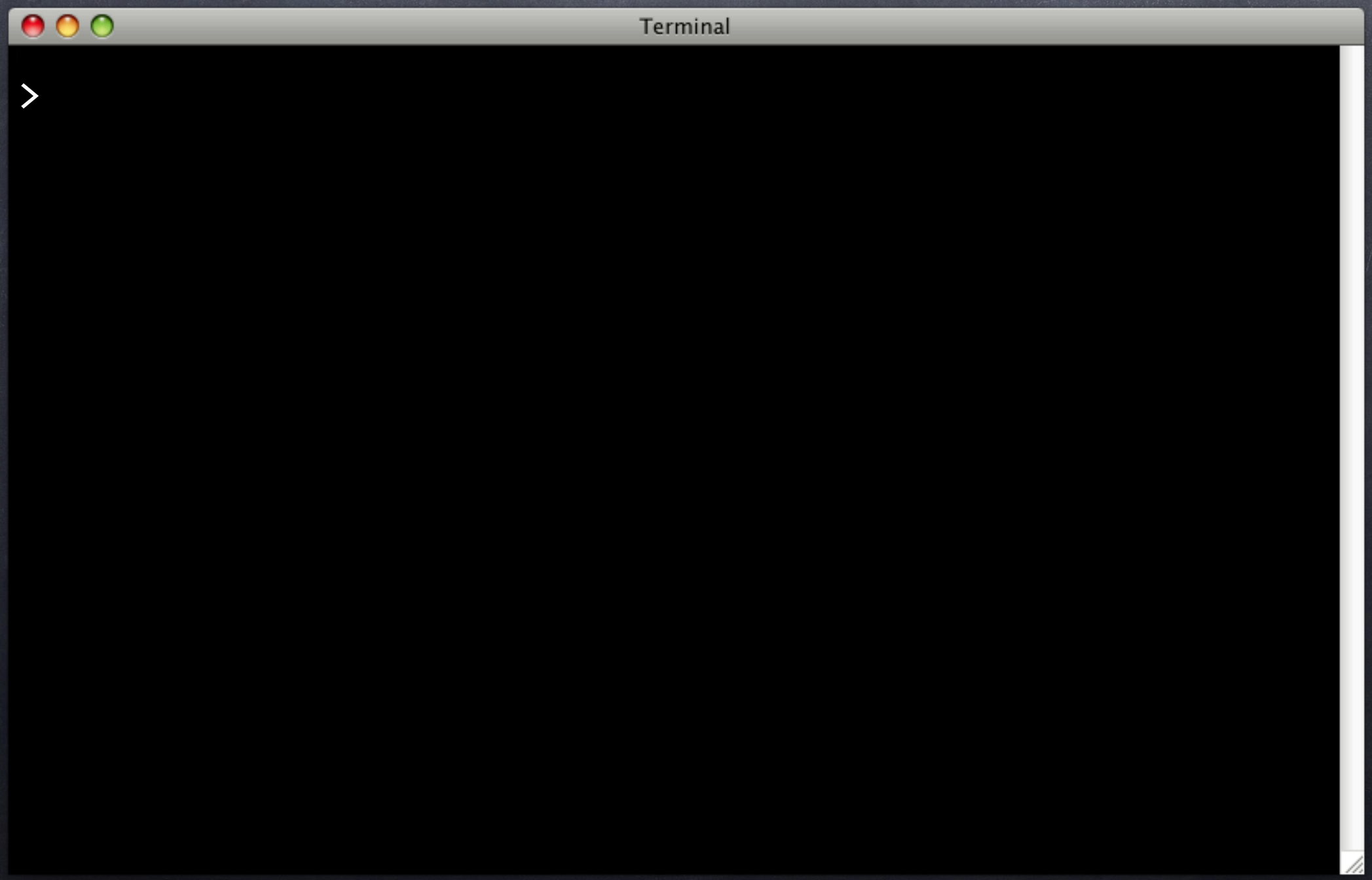


Verify Users

Terminal

```
> sqitch verify
Verifying flipr_test
  * appschema .. ok
  * users ...... ok
Verify successful
>
```

Status Update



Status Update

Terminal

```
> sqitch status
# On database flipr_test
# Project:  flipr
# Change:   bbb98e0660898ced0ba5f84c577036b5dafa3a55
# Name:     users
# Deployed: 2012-12-31 11:42:17 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

Status Update

Terminal

```
> sqitch status
# On database flipr_test
# Project:  flipr
# Change:   bbb98e0660898ced0ba5f84c577036b5dafa3a55
# Name:     users
# Deployed: 2012-12-31 11:42:17 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

Status Update

Terminal

```
> sqitch status
# On database flipr_test
# Project:  flipr
# Change:   bbb98e0660898ced0ba5f84c577036b5dafa3a55
# Name:     users
# Deployed: 2012-12-31 11:42:17 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
> sqitch revert --to @HEAD^ -y
Reverting changes to appschema from flipr_test
  - users .. ok
>
```

Status Update

Terminal

```
> sqitch status
# On database flipr_test
# Project:  flipr
# Change:   bbb98e0660898ced0ba5f84c577036b5dafa3a55
# Name:     users
# Deployed: 2012-12-31 11:42:17 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
> sqitch revert --to @HEAD^ -y
Reverting changes to appschema from flipr_test
  - users .. ok
>
```

Yes, really.

Status Update

Terminal

```
> sqitch status
# On database flipr_test
# Project:  flipr
# Change:   bbb98e0660898ced0ba5f84c577036b5dafa3a55
# Name:     users
# Deployed: 2012-12-31 11:42:17 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
> sqitch revert --to @HEAD^ -y
Reverting changes to appschema from flipr_test
  - users .. ok
>
```

Symbolic Tags

Symbolic Tags

- @HEAD Last change

Symbolic Tags

- @HEAD Last change
- @ROOT First change

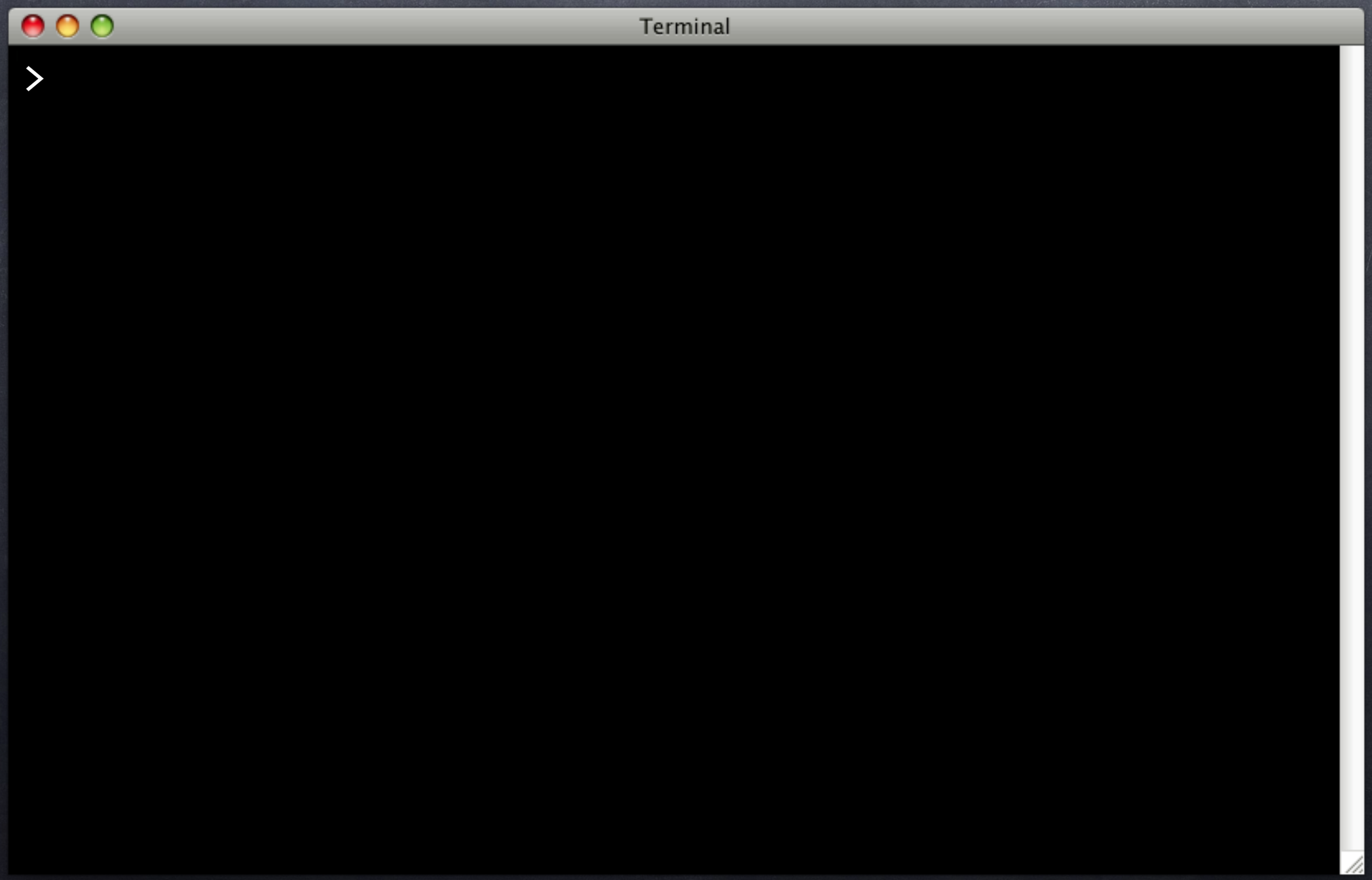
Symbolic Tags

- @HEAD Last change
- @ROOT First change
- ^ Previous change

Symbolic Tags

- @HEAD Last change
- @ROOT First change
- ^ Previous change
- ~ Following change

Lose Users



Lose Users

Terminal

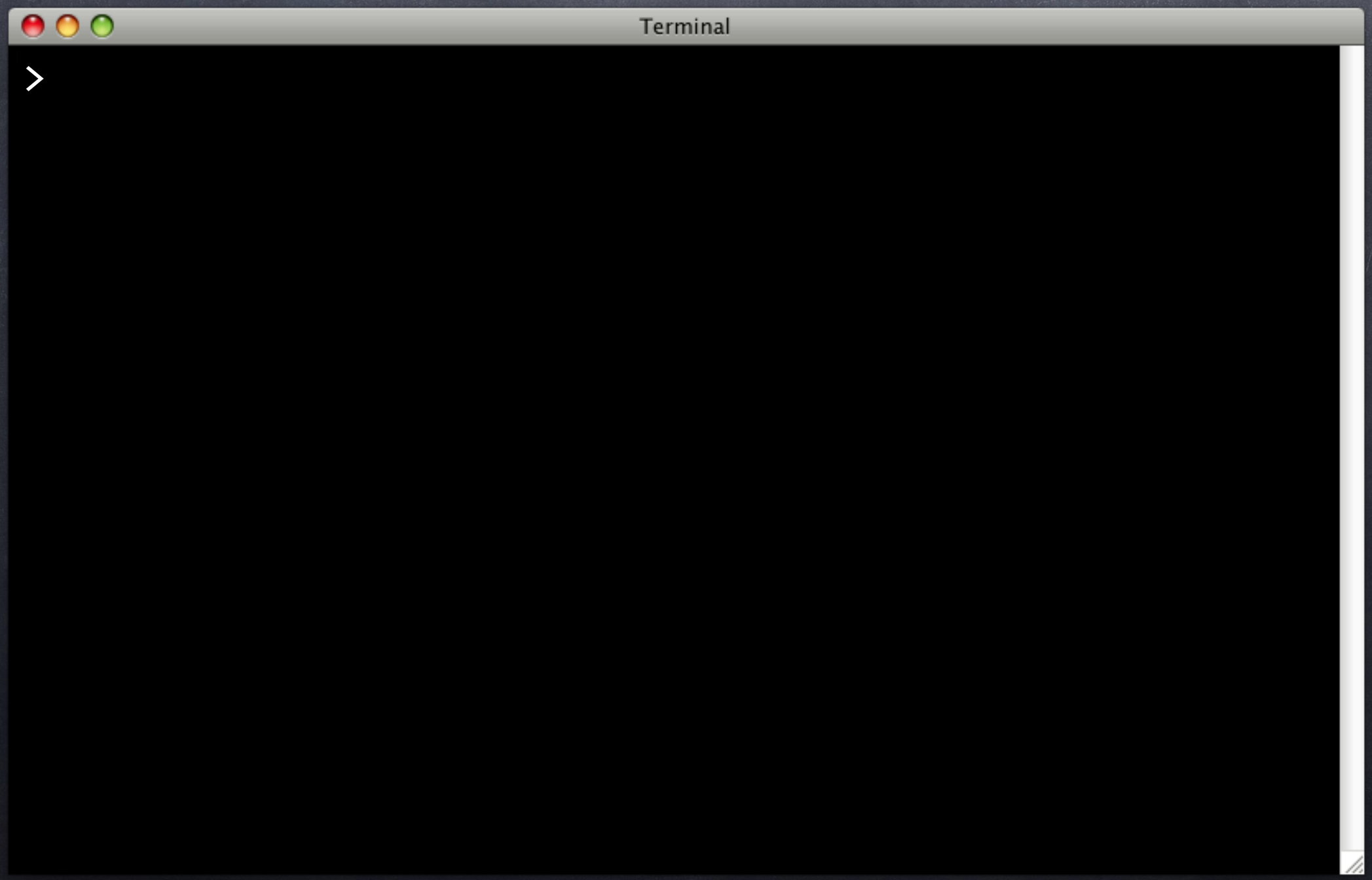
```
> psql -d flipr_test -c '\d flipr.users'
Did not find any relation named "flipr.users".
>
```

Lose Users

Terminal

```
> psql -d flipr_test -c '\d flipr.users'
Did not find any relation named "flipr.users".
> sqitch status
# On database flipr_test
# Project:  flipr
# Change:   ae1263b92b3355cf705c7832271730fe4ccb3c77
# Name:     appschema
# Deployed: 2012-12-31 11:03:18 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Undeployed change:
  * users
>
```

Lose Users

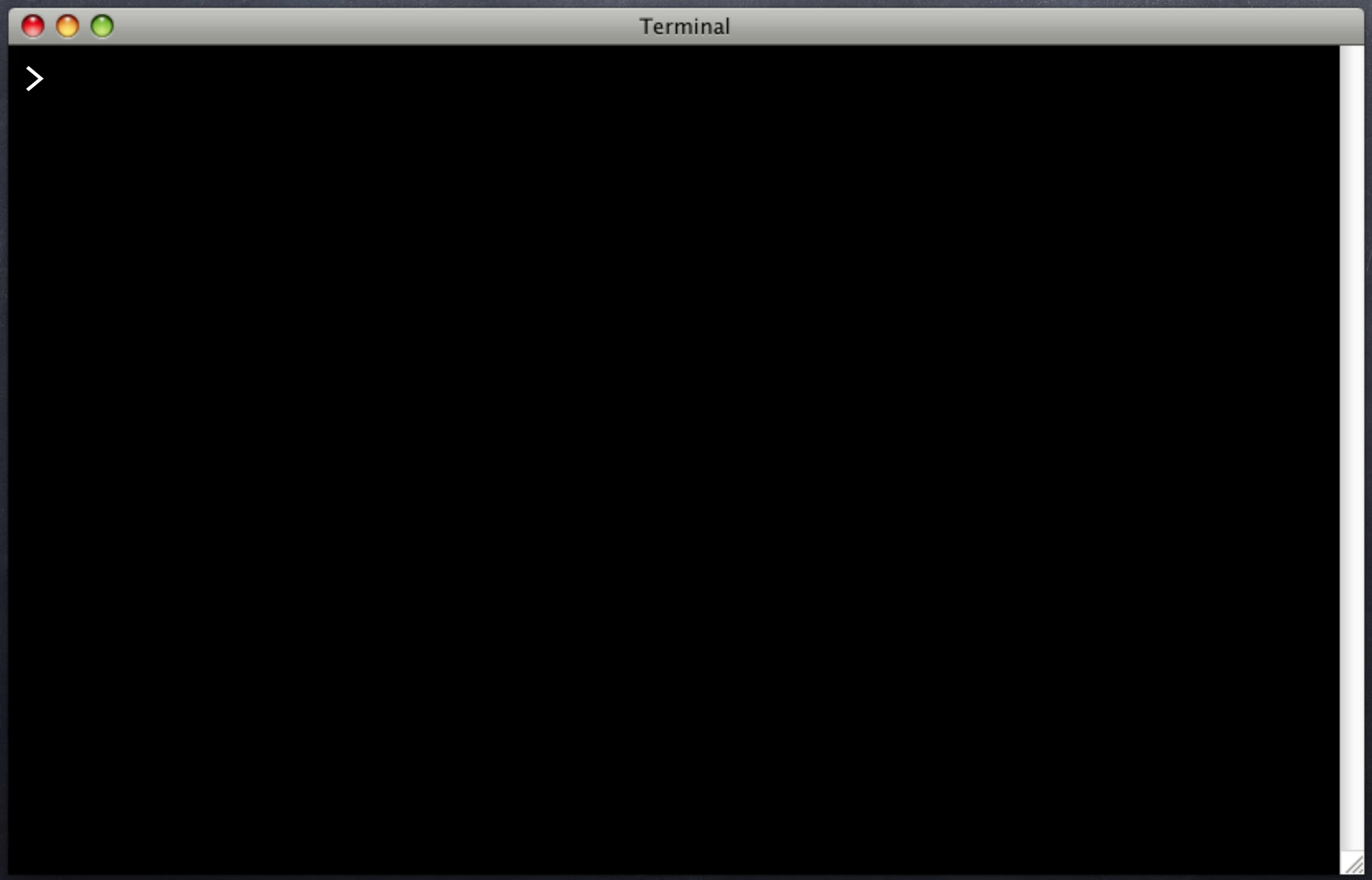


Lose Users

Terminal

```
> sqitch verify
Verifying flipr_test
  * appschema .. ok
Undeployed change:
  * users
Verify successful
>
```

Lose Users



Lose Users

Terminal

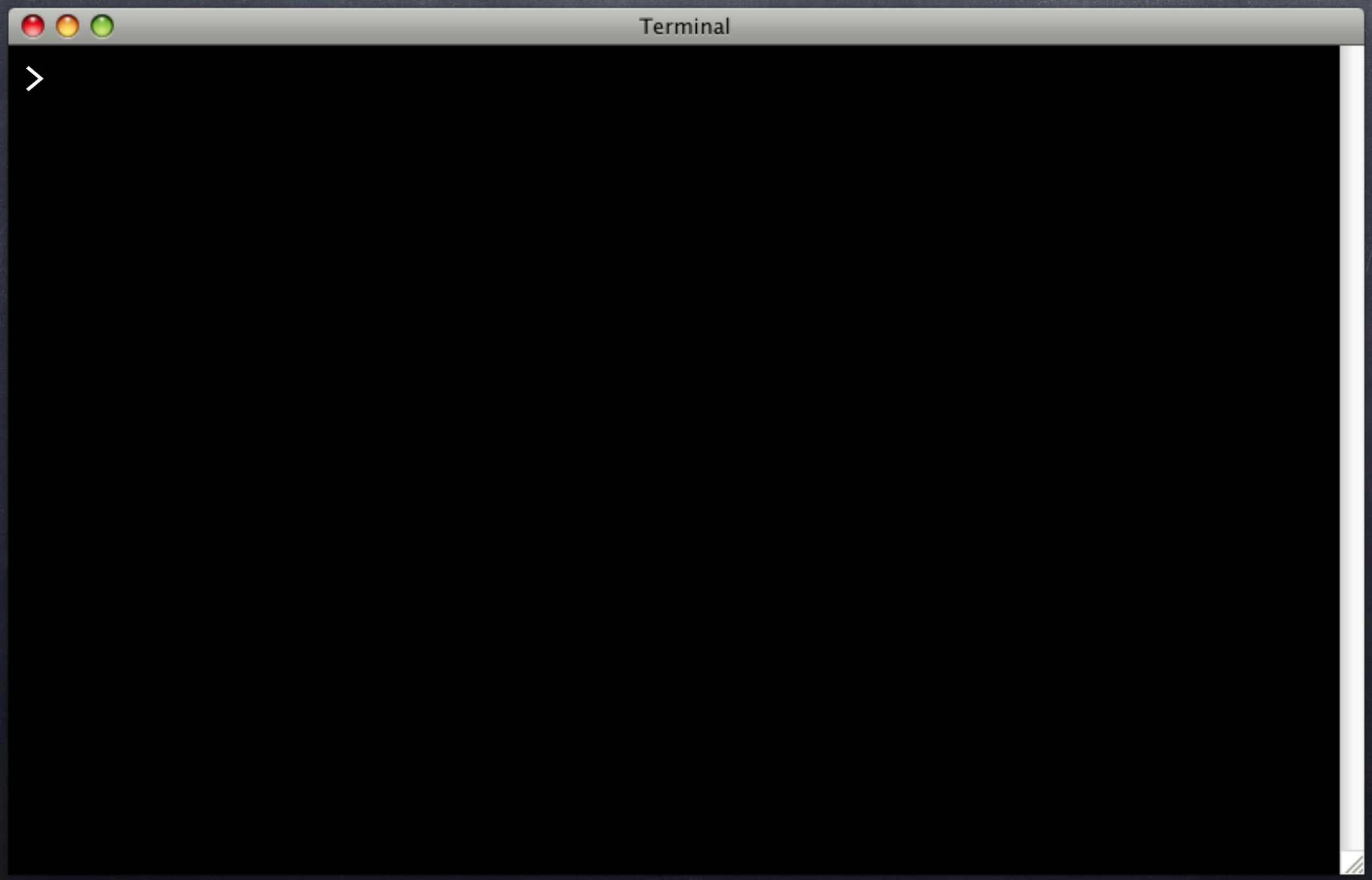
```
> git add .
> git commit -am 'Add users table.'
[master 6aae0e9] Add users table.
 4 files changed, 31 insertions(+)
 create mode 100644 deploy/users.sql
 create mode 100644 revert/users.sql
 create mode 100644 verify/users.sql
>
```

Lose Users

Terminal

```
> git add .
> git commit -am 'Add users table.'
[master 6aae0e9] Add users table.
 4 files changed, 31 insertions(+)
 create mode 100644 deploy/users.sql
 create mode 100644 revert/users.sql
 create mode 100644 verify/users.sql
> sqitch deploy
Deploying changes to flipr_test
  + users .. ok
>
```

Up to Date



Up to Date

Terminal

```
> sqitch status
# On database flipr_test
# Project:  flipr
# Change:   bbb98e0660898ced0ba5f84c577036b5dafa3a55
# Name:     users
# Deployed: 2012-12-31 14:04:11 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

Up to Date

Terminal

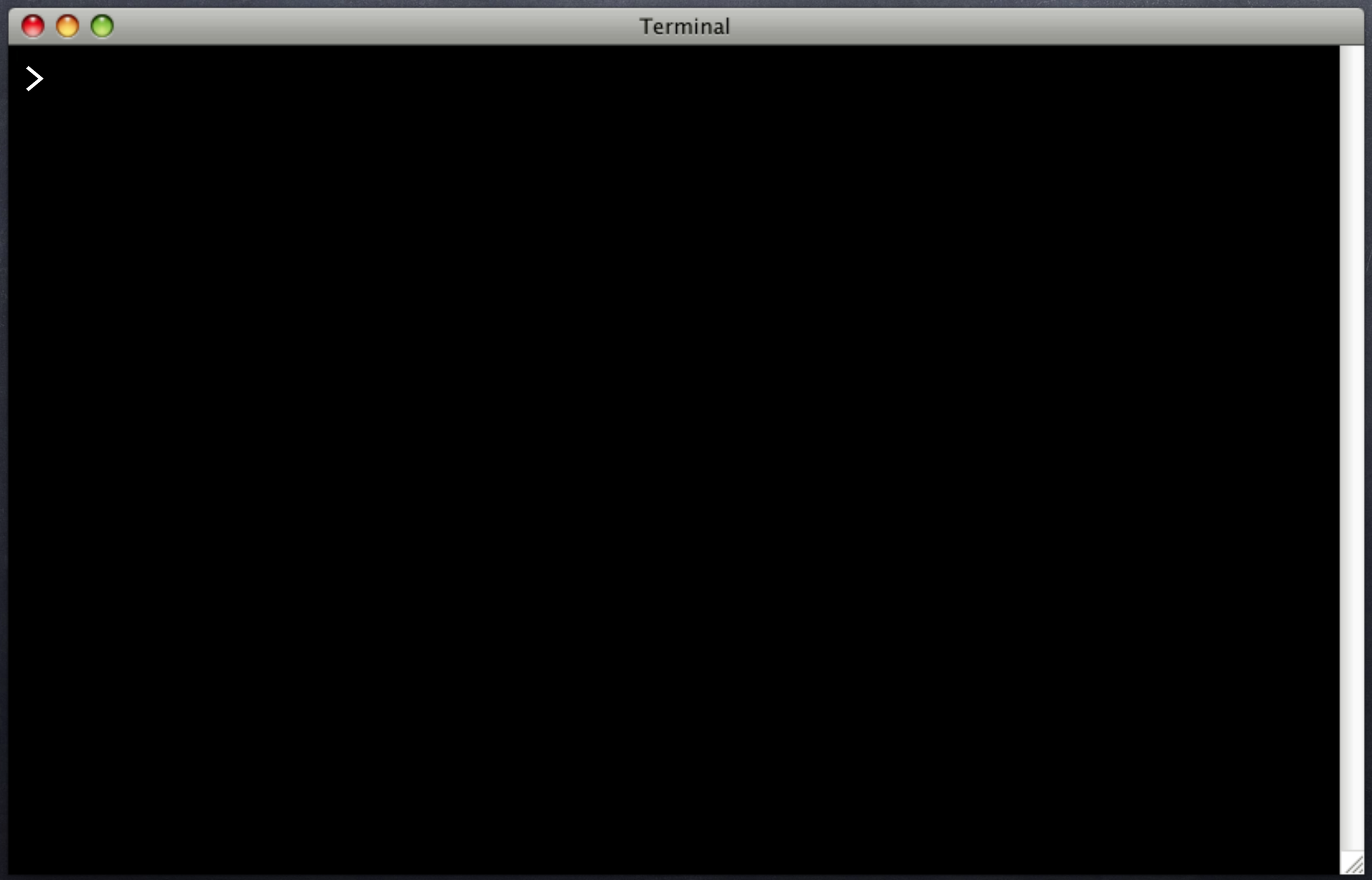
```
> sqitch status
# On database flipr_test
# Project:  flipr
# Change:   bbb98e0660898ced0ba5f84c577036b5dafa3a55
# Name:     users
# Deployed: 2012-12-31 14:04:11 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

Up to Date

Terminal

```
> sqitch status
# On database flipr_test
# Project:  flipr
# Change:   bbb98e0660898ced0ba5f84c577036b5dafa3a55
# Name:     users
# Deployed: 2012-12-31 14:04:11 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

A Twofer



A Twofer

Terminal

```
> sqitch add insert_user -r users -r appschema \
  -n 'Creates a function to insert a user.'
Created deploy/insert_user.sql
Created revert/insert_user.sql
Created verify/insert_user.sql
Added "insert_user [users appschema]" to sqitch.plan

>
```

A Twofer

Terminal

```
> sqitch add insert_user -r users -r appschema \
  -n 'Creates a function to insert a user.'
Created deploy/insert_user.sql
Created revert/insert_user.sql
Created verify/insert_user.sql
Added "insert_user [users appschema]" to sqitch.plan

>
```

A Twofer

Terminal

```
> sqitch add insert_user -r users -r appschema \
  -n 'Creates a function to insert a user.'
Created deploy/insert_user.sql
Created revert/insert_user.sql
Created verify/insert_user.sql
Added "insert_user [users appschema]" to sqitch.plan

> sqitch add change_pass -r users -r appschema \
  -n 'Creates a function to change a user password.'
Created deploy/change_pass.sql
Created revert/change_pass.sql
Created verify/change_pass.sql
Added "change_pass [users appschema]" to sqitch.plan

>
```

A Twofer

Terminal

```
> sqitch add insert_user -r users -r appschema \
  -n 'Creates a function to insert a user.'
Created deploy/insert_user.sql
Created revert/insert_user.sql
Created verify/insert_user.sql
Added "insert_user [users appschema]" to sqitch.plan

> sqitch add change_pass -r users -r appschema \
  -n 'Creates a function to change a user password.'
Created deploy/change_pass.sql
Created revert/change_pass.sql
Created verify/change_pass.sql
Added "change_pass [users appschema]" to sqitch.plan

>
```

A Twofer

Terminal

```
> sqitch add insert_user -r users -r appschema \
  -n 'Creates a function to insert a user.'
Created deploy/insert_user.sql
Created revert/insert_user.sql
Created verify/insert_user.sql
Added "insert_user [users appschema]" to sqitch.plan

> sqitch add change_pass -r users -r appschema \
  -n 'Creates a function to change a user password.'
Created deploy/change_pass.sql
Created revert/change_pass.sql
Created verify/change_pass.sql
Added "change_pass [users appschema]" to sqitch.plan

> emacs sqitch.plan
```

sqitch.plan

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
```

--:-- sqitch.plan All (SQL[ansi])

sqitch.plan

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
```

--:-- sqitch.plan All (SQL[ansi])

sqitch.plan

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
```

--:-- sqitch.plan All (SQL[ansi])

sqitch.plan

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
```

--:-- sqitch.plan All (SQL[ansi])

sqitch.plan

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
```

--:-- sqitch.plan All (SQL[ansi])

sqitch.plan

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
```

--:-- sqitch.plan All (SQL[ansi])

sqitch.plan

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
```

--:-- sqitch.plan All (SQL[ansi])

sqitch.plan

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
```

--:-- sqitch.plan All (SQL[ansi])

A Twofer

Terminal

```
> emacs sqitch.plan
>
```

A Twofer

Terminal

```
> emacs sqitch.plan
> emacs deploy/insert_user.sql
```

deploy/insert_user.sql

Emacs

```
-- Deploy insert_user
-- requires: appuser
-- requires: users

BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

--:-- deploy/insert_u All (SQL[ansi])

deploy/insert_user.sql

Emacs

```
-- Deploy insert_user
-- requires: appuser
-- requires: users

BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

--:-- deploy/insert_u All (SQL[ansi])

deploy/insert_user.sql

Emacs

```
-- Deploy insert_user
-- requires: appuser
-- requires: users

BEGIN;

CREATE OR REPLACE FUNCTION flipr.insert_user(
    nickname TEXT,
    password TEXT
) RETURNS VOID LANGUAGE SQL SECURITY DEFINER AS $$
    INSERT INTO users VALUES($1, md5($2));
$$;

COMMIT;
```

--:-- deploy/insert_u All (SQL[ansi])

A Twofer

Terminal

```
> emacs sqitch.plan
> emacs deploy/insert_user.sql
>
```

A Twofer

Terminal

```
> emacs sqitch.plan
> emacs deploy/insert_user.sql
> emacs verify/insert_user.sql
```

verify/insert_user.sql

Emacs

```
-- Verify insert_user

BEGIN;

-- XXX Add verifications here.

COMMIT;
```

--:-- verify/insert_u All (SQL[ansi])

verify/insert_user.sql

Emacs

```sql
-- Verify insert_user

BEGIN;

SELECT has_function_privilege(
    'flipr.insert_user(text, text)',
    'execute'
);

COMMIT;
```

--:-- verify/insert_u All (SQL[ansi])

A Twofer

Terminal

```
> emacs sqitch.plan
> emacs deploy/insert_user.sql
> emacs verify/insert_user.sql
>
```

A Twofer

Terminal

```
> emacs sqitch.plan
> emacs deploy/insert_user.sql
> emacs verify/insert_user.sql
> emacs insert/change_pass.sql
```

deploy/change_pass.sql

Emacs

```
-- Deploy change_pass
-- :requires: appuser
-- :requires: users

BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

--:-- deploy/change_p All (SQL[ansi])

deploy/change_pass.sql

Emacs

```sql
-- Deploy change_pass
-- :requires: appuser
-- :requires: users

BEGIN;
CREATE OR REPLACE FUNCTION flipr.change_pass(
      nick TEXT, oldpass TEXT, newpass TEXT
) RETURNS BOOLEAN LANGUAGE plpgsql SECURITY DEFINER AS $$
BEGIN
    UPDATE users SET password = md5($3)
     WHERE nickname = $1 AND password = md5($2);
    RETURN FOUND;
END;
$$;

COMMIT;
```

--:-- deploy/change_p All (SQL[ansi])

A Twofer

Terminal

```
> emacs sqitch.plan
> emacs deploy/insert_user.sql
> emacs verify/insert_user.sql
> emacs insert/change_pass.sql
>
```

A Twofer

Terminal

```
> emacs sqitch.plan
> emacs deploy/insert_user.sql
> emacs verify/insert_user.sql
> emacs insert/change_pass.sql
> emacs verify/change_pass.sql
```

verify/change_pass.sql

Emacs

```
-- Verify change_pass

BEGIN;

-- XXX Add verifications here.

COMMIT;
```

--:-- verify/change_p All (SQL[ansi])

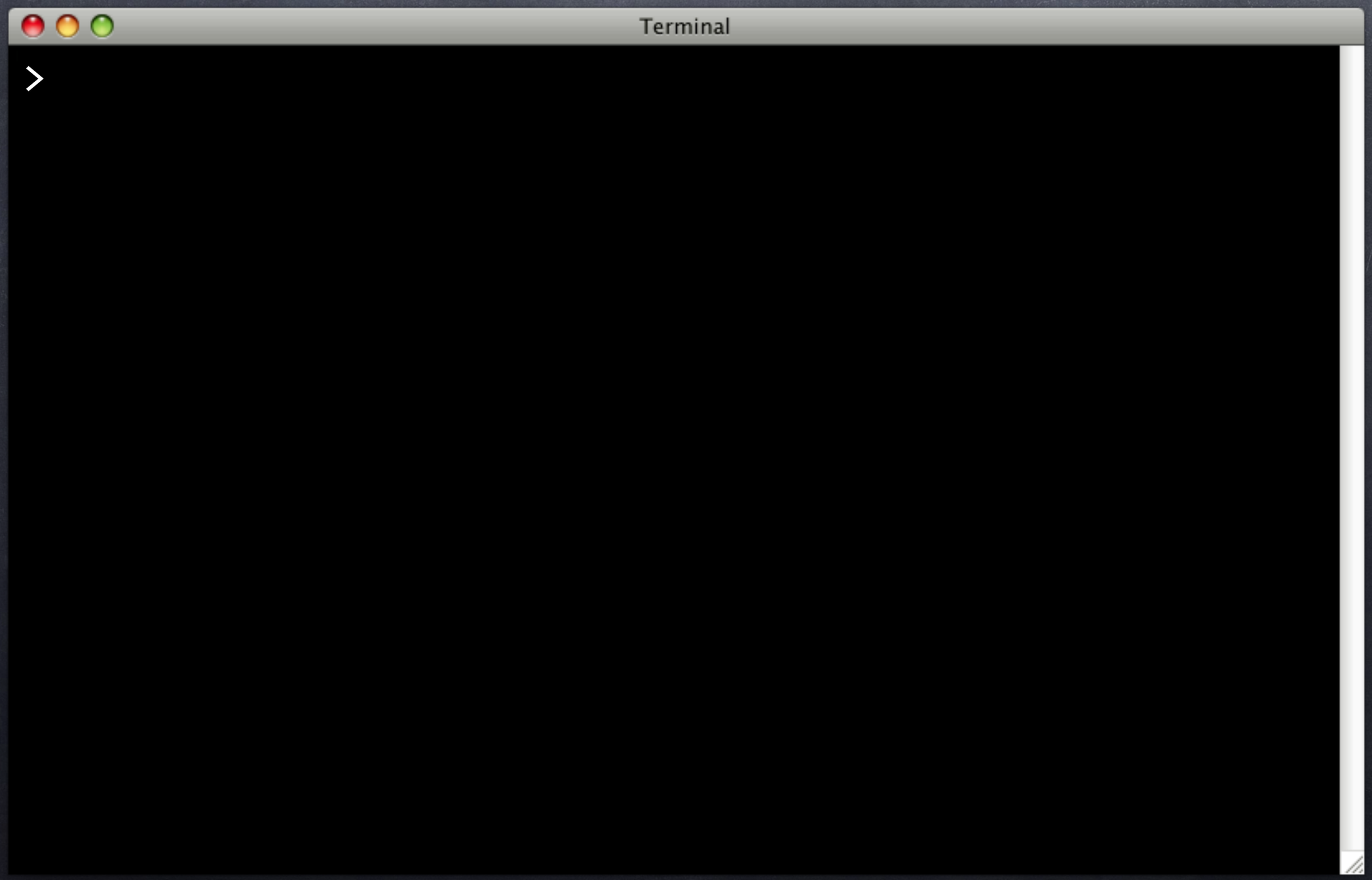
verify/change_pass.sql

Emacs

```sql
-- Verify change_pass

BEGIN;

SELECT has_function_privilege(
    'flipr.change_pass(text, text, text)',
    'execute'
);

COMMIT;
```

--:-- verify/change_p All (SQL[ansi])

Deploy Functions



Deploy Functions

Terminal

```
> sqitch deploy
Deploying changes to flipr_test
  + insert_user .. ok
  + change_pass .. ok
>
```

Deploy Functions

Terminal

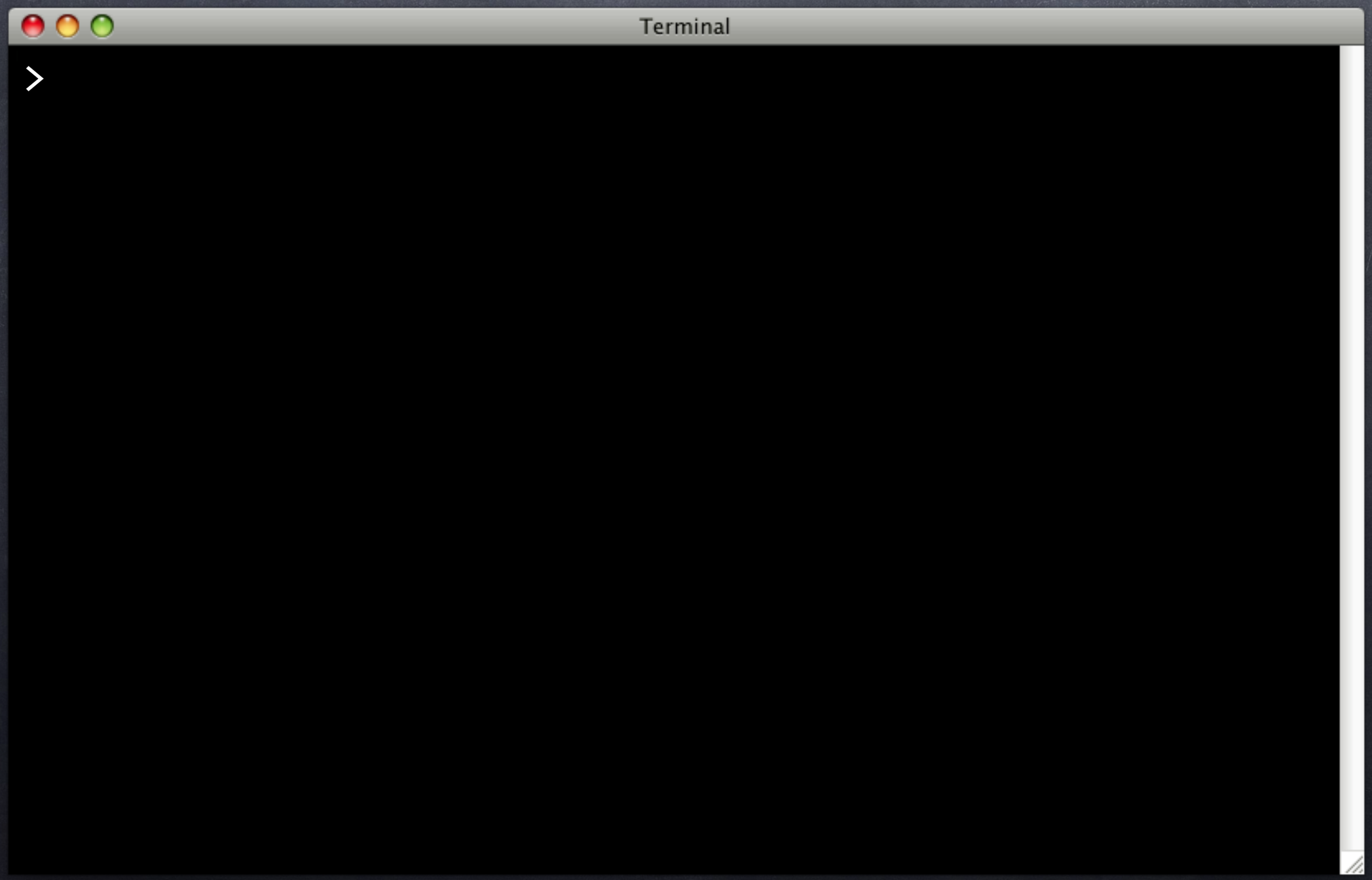
```
> sqitch deploy
Deploying changes to flipr_test
  + insert_user .. ok
  + change_pass .. ok
> psql -d flipr_test -c '\df flipr.*'
                                   List of function
 Schema |     Name      | Result data type |
--------+---------------+------------------+-----------
 flipr  | change_pass | boolean          | nick text
 flipr  | insert_user | void             | nickname
>
```

Deploy Functions

Terminal

```
> sqitch deploy
Deploying changes to flipr_test
  + insert_user .. ok
  + change_pass .. ok
> psql -d flipr_test -c '\df flipr.*'
                                    List of function
 Schema |     Name      | Result data type |
--------+---------------+------------------+-----------
 flipr  | change_pass   | boolean          | nick text
 flipr  | insert_user   | void             | nickname
>
```

Status Update

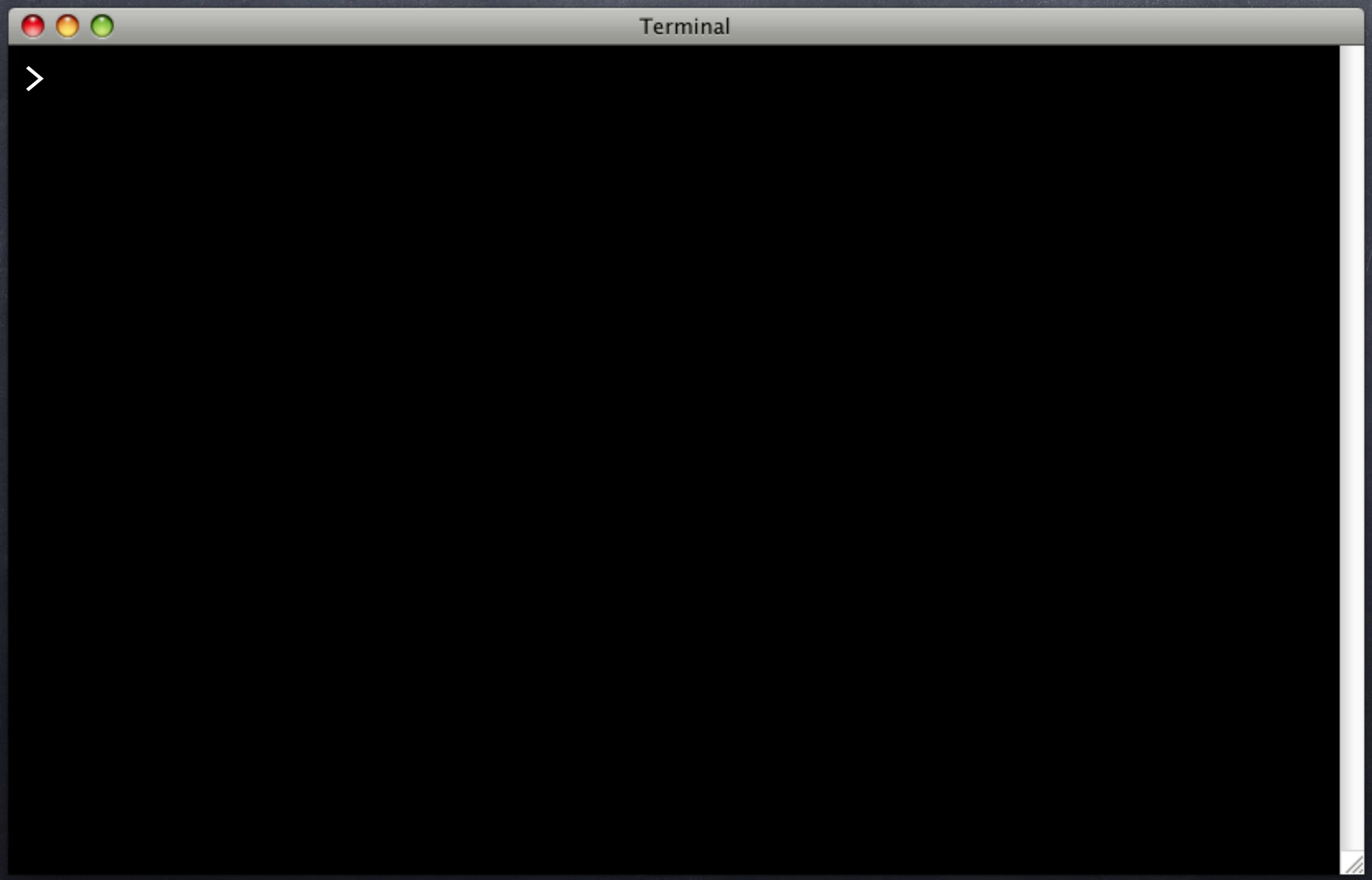


Status Update

Terminal

```
> sqitch status
# On database flipr_test
# Project:  flipr
# Change:   0acf77aaafc1a828e37e8e857d04d13ee22a2d36
# Name:     change_pass
# Deployed: 2012-12-31 14:45:18 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

Revert



Revert

Terminal

```
> sqitch revert -y --to @HEAD^^
Reverting changes to users from flipr_test
  - change_pass .. ok
  - insert_user .. ok
>
```

Revert

Terminal

```
> sqitch revert -y --to @HEAD^^
Reverting changes to users from flipr_test
  - change_pass .. ok
  - insert_user .. ok
>
```

Revert

Terminal

```
> sqitch revert -y --to @HEAD^^
Reverting changes to users from flipr_test
  - change_pass .. ok
  - insert_user .. ok
> psql -d flipr_test -c '\df flipr.*'
                        List of functions
 Schema | Name | Result data type | Argument data ty
--------+------+------------------+-----------------
>
```

Commit It

Commit It

- git add && git commit

Commit It

- git add && git commit
- sqitch deploy

Commit It

- git add && git commit
- sqitch deploy
- And now...

Commit It

- git add && git commit
- sqitch deploy
- And now...
  - Tag

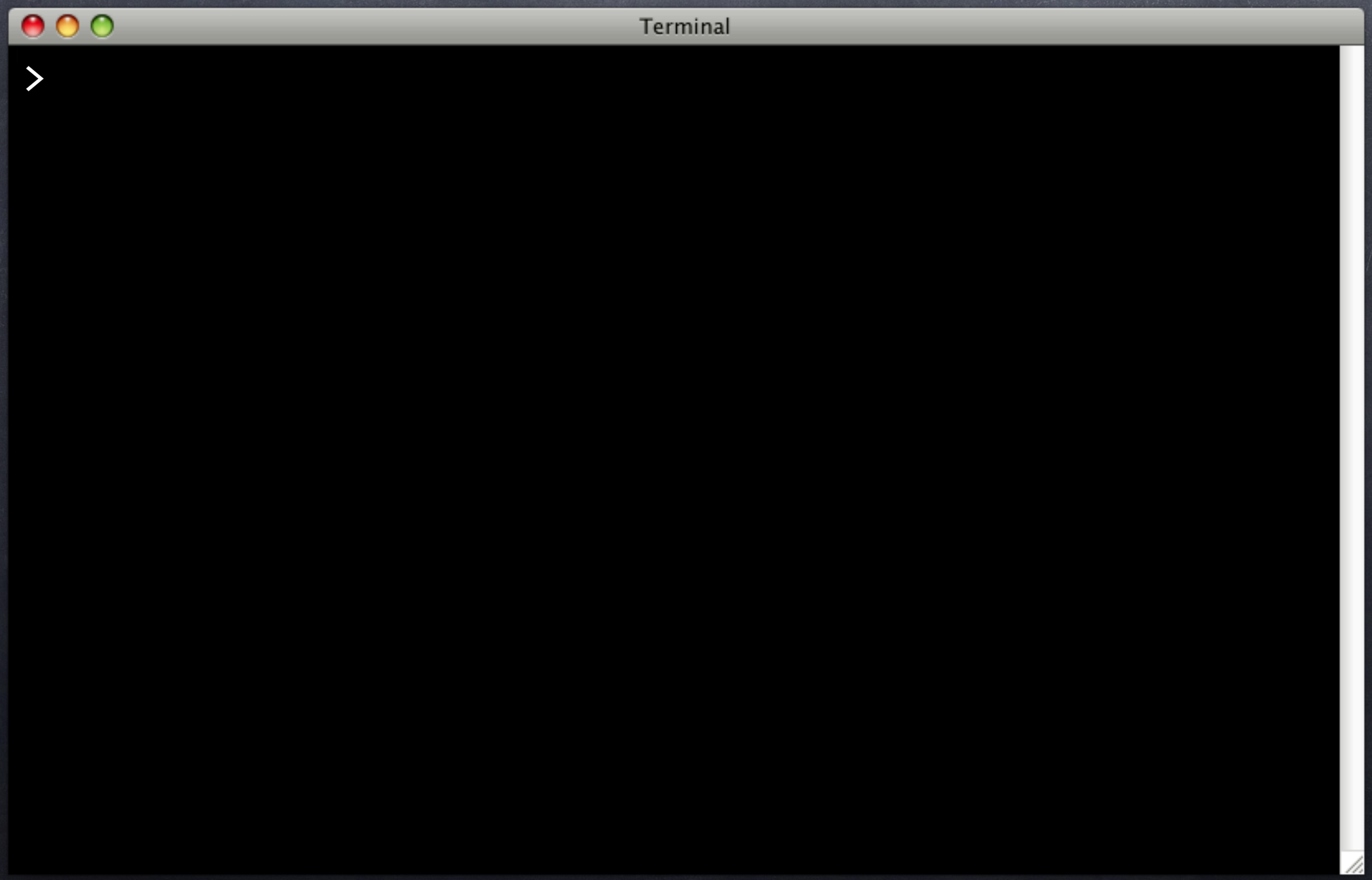
Commit It

- git add && git commit
- sqitch deploy
- And now...
  - Tag
  - Bundle

Commit It

- git add && git commit
- sqitch deploy
- And now...
  - Tag
  - Bundle
  - Ship!

Tag It!



Tag It!

Terminal

```
> sqitch tag v1.0.0-dev1 -n 'Tag v1.0.0-dev1.'
Tagged "change_pass" with @v1.0.0-dev1
>
```

Tag It!

Terminal

```
> sqitch tag v1.0.0-dev1 -n 'Tag v1.0.0-dev1.'
Tagged "change_pass" with @v1.0.0-dev1
>
```

Tag It!

Terminal

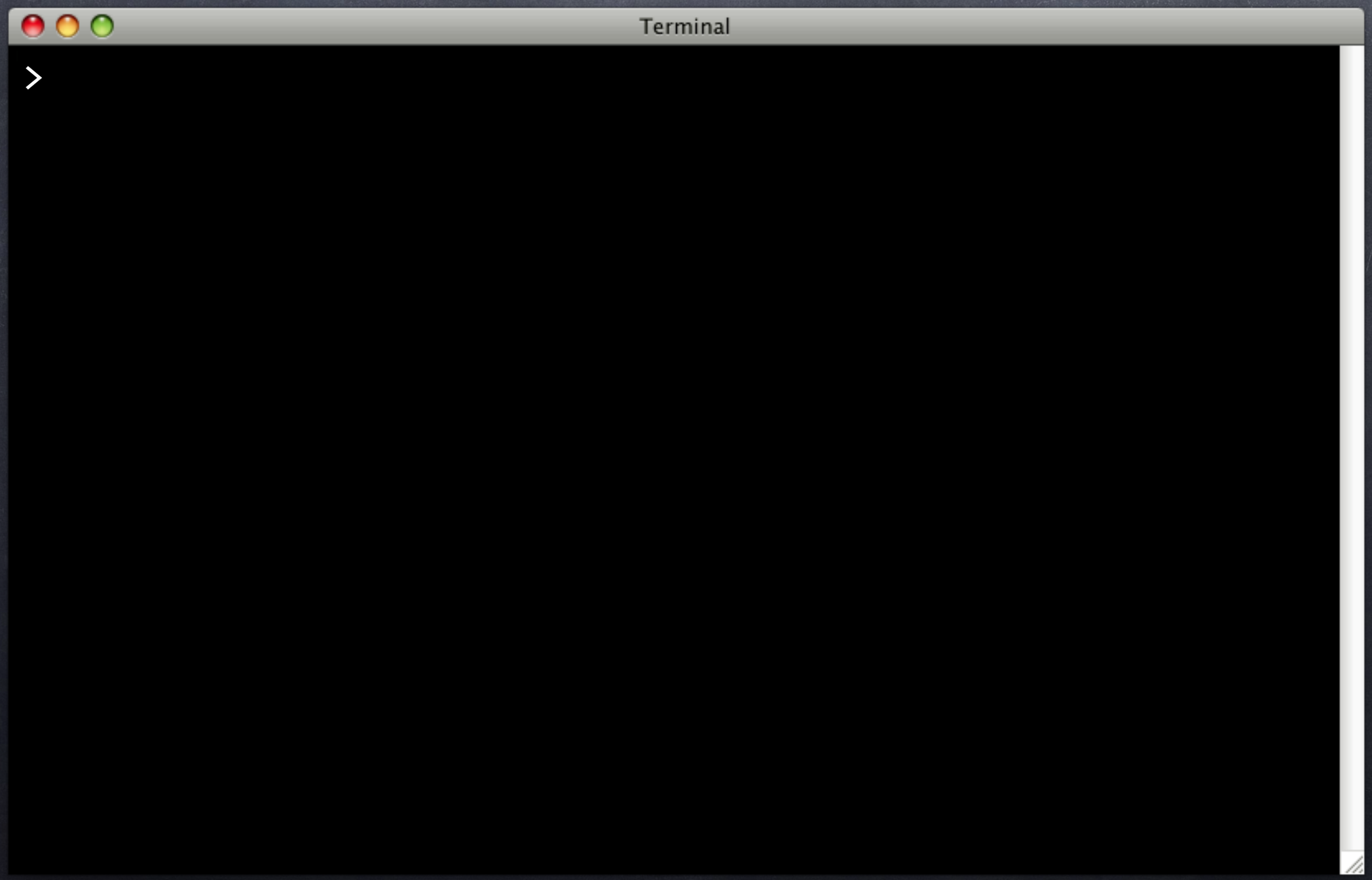
```
> sqitch tag v1.0.0-dev1 -n 'Tag v1.0.0-dev1.'
Tagged "change_pass" with @v1.0.0-dev1
> git commit -am 'Tag the database with v1.0.0-dev1.
[master b215e03] Tag the database with v1.0.0-dev1.
 1 file changed, 1 insertion(+)
>
```

Tag It!

Terminal

```
> sqitch tag v1.0.0-dev1 -n 'Tag v1.0.0-dev1.'
Tagged "change_pass" with @v1.0.0-dev1
> git commit -am 'Tag the database with v1.0.0-dev1.
[master b215e03] Tag the database with v1.0.0-dev1.
 1 file changed, 1 insertion(+)
> git tag v1.0.0-dev1 -am 'Tag v1.0.0-dev1'
>
```

Bundle It!



Bundle It!

Terminal

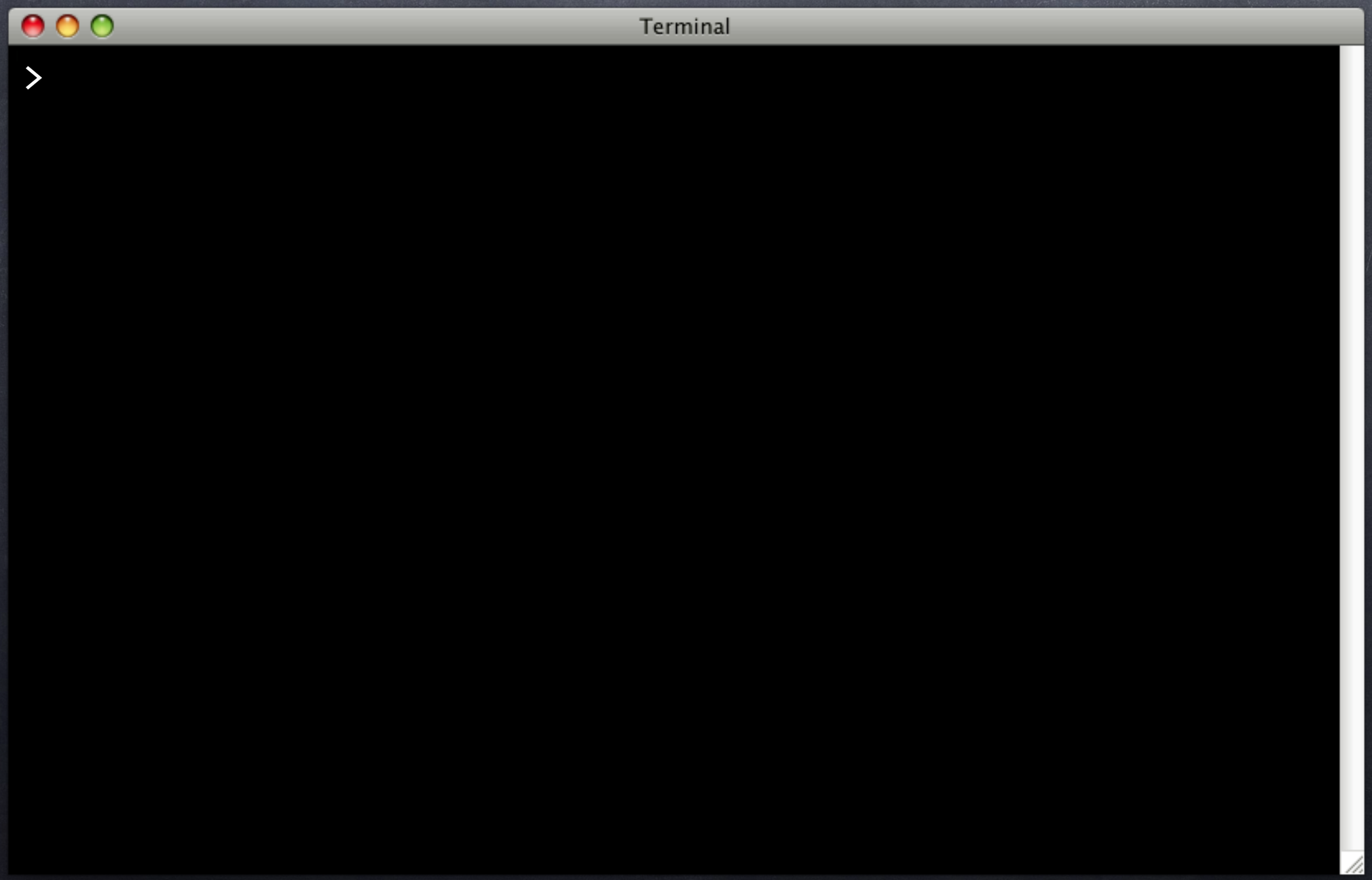
```
> sqitch bundle
Bundling into bundle/
Writing config
Writing plan
Writing scripts
  + appschema
  + users
  + insert_user
  + change_pass @v1.0.0-dev1
>
```

Bundle It!

Terminal

```
> sqitch bundle
Bundling into bundle/
Writing config
Writing plan
Writing scripts
  + appschema
  + users
  + insert_user
  + change_pass @v1.0.0-dev1
>
```

Test and Ship It!



Test and Ship It!

Terminal

```
> cd bundle
>
```

Test and Ship It!

Terminal

```
> cd bundle
> createdb flipr_stage
>
```

Test and Ship It!

Terminal

```
> cd bundle
> createdb flipr_stage
> sqitch --db-name flipr_stage deploy
Adding metadata tables to flipr_stage
Deploying changes to flipr_stage
  + appschema ................... ok
  + users ....................... ok
  + insert_user ................. ok
  + change_pass @v1.0.0-dev1 .. ok
>
```

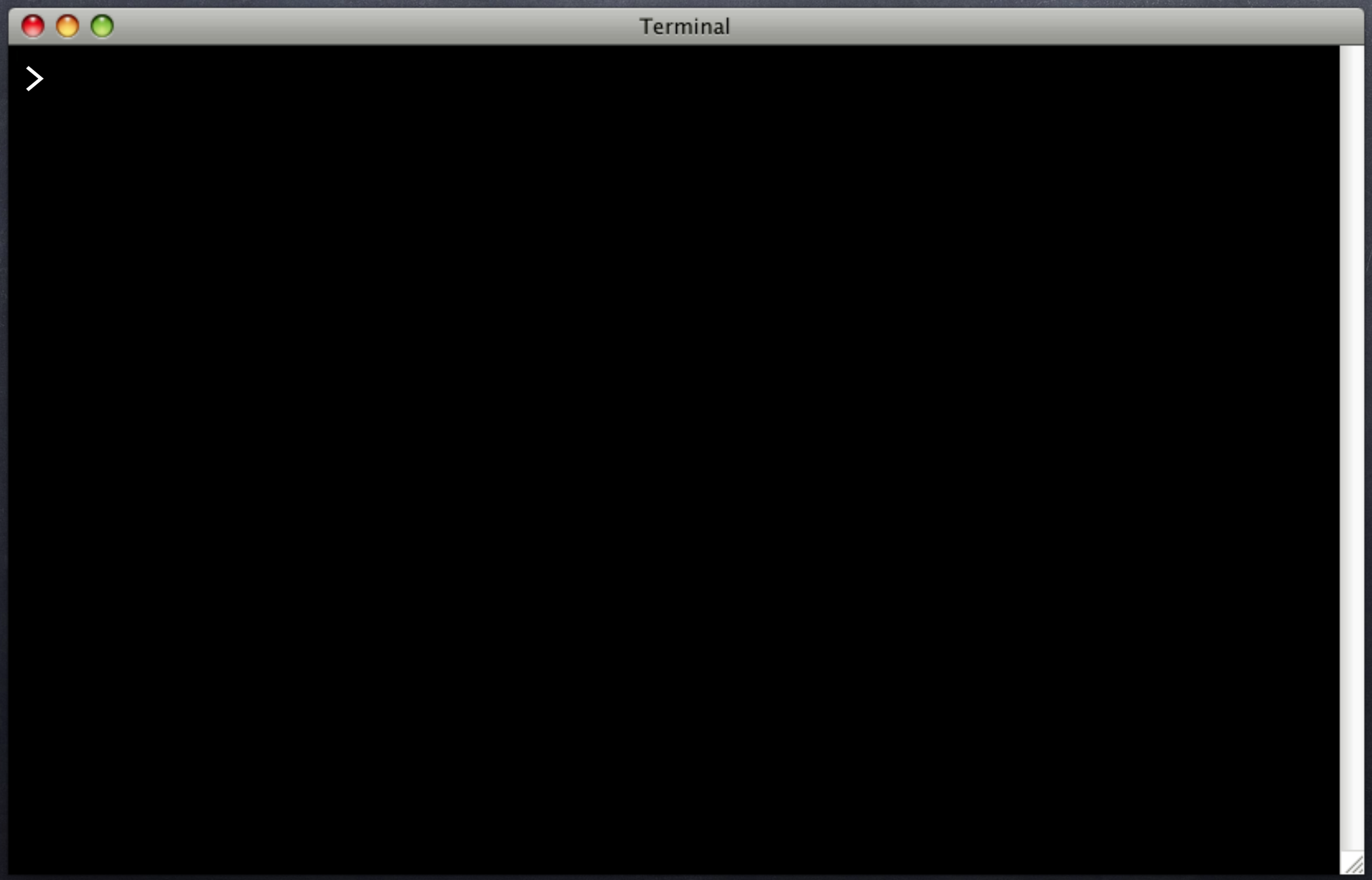
Test and Ship It!

Terminal

```
> cd bundle
> createdb flipr_stage
> sqitch --db-name flipr_stage deploy
Adding metadata tables to flipr_stage
Deploying changes to flipr_stage
  + appschema ................... ok
  + users ....................... ok
  + insert_user ................. ok
  + change_pass @v1.0.0-dev1 .. ok
> cd ..
> mv bundle flipr-1.0.0-dev1
> tar -czf flipr-1.0.0-dev1.tar.gz flipr-1.0.0-dev1
>
```

Reworking Changes

Ruh-Roh



Ruh-Roh

Terminal

```
> psql -d flipr_test -c "
    SELECT insert_user('foo', 'secr3t'),
           insert_user('bar', 'secr3t');
    SELECT nickname, password FROM users;
"
 nickname |                 password
----------+----------------------------------
 foo      | 9695da4dd567a19f9b92065f240c6725
 bar      | 9695da4dd567a19f9b92065f240c6725
```

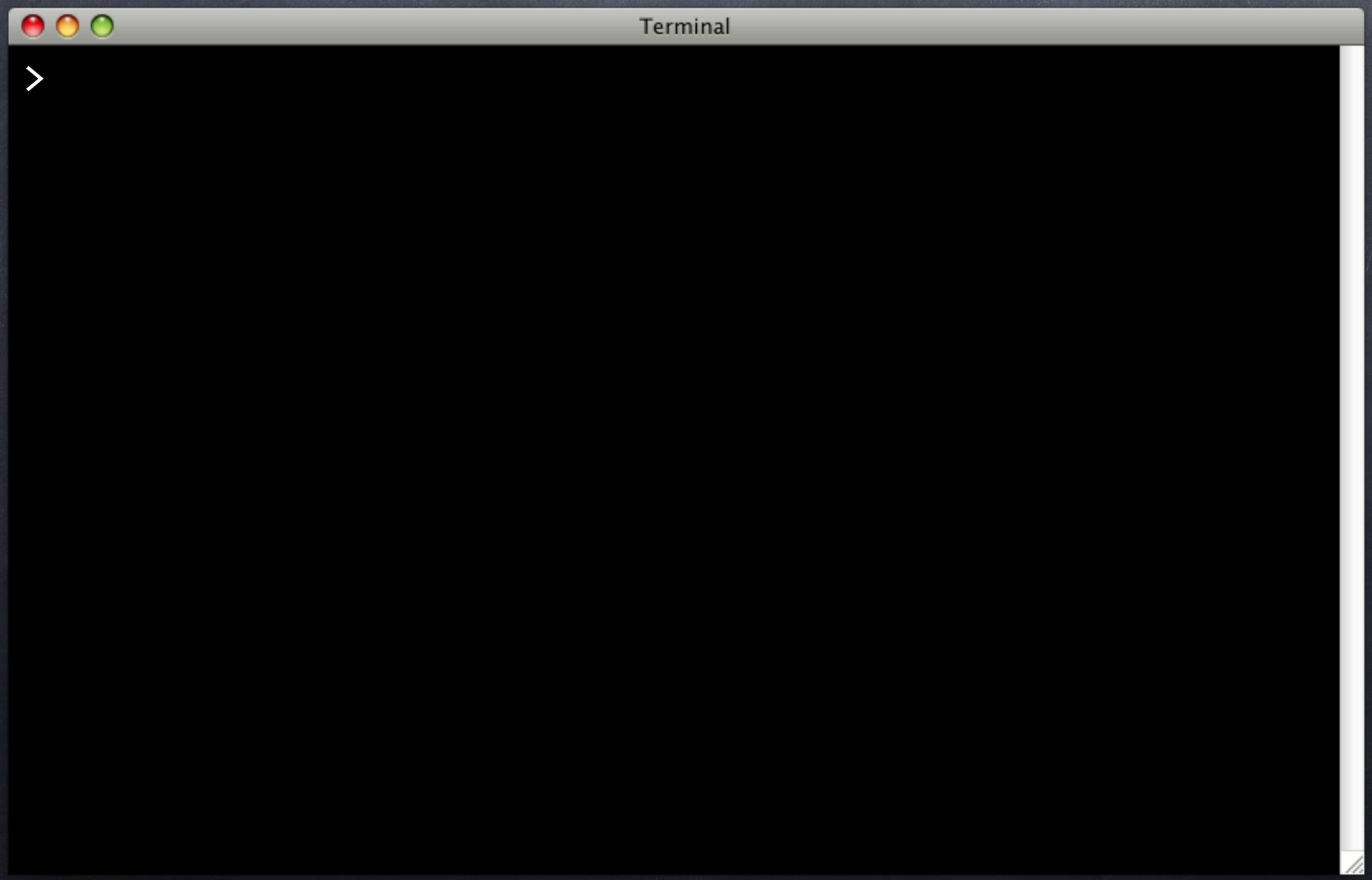
Ruh-Roh

Terminal

```
> psql -d flipr_test -c "
    SELECT insert_user('foo', 'secr3t'),
           insert_user('bar', 'secr3t');
    SELECT nickname, password FROM users;
"
 nickname |                 password
----------+----------------------------------
 foo      | 9695da4dd567a19f9b92065f240c6725
 bar      | 9695da4dd567a19f9b92065f240c6725
```

Not good.

Add pgcrypto



Add pgcrypto

Terminal

```
> sqitch add pgcrypto -n 'Loads pgcrypto extension.'
Adding deploy/pgcrypto.sql
Adding revert/pgcrypto.sql
Adding verify/pgcrypto.sql
Added "pgcrypto" to sqitch.plan
>
```

Add pgcrypto

Terminal

```
> sqitch add pgcrypto -n 'Loads pgcrypto extension.'
Adding deploy/pgcrypto.sql
Adding revert/pgcrypto.sql
Adding verify/pgcrypto.sql
Added "pgcrypto" to sqitch.plan
> emacs deploy/pgcrpyto.sql
```

deploy/pgcrypto.sql

Emacs

```
-- Deploy pgcrypto

BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

--:-- deploy/pgcrypto All (SQL[ansi])

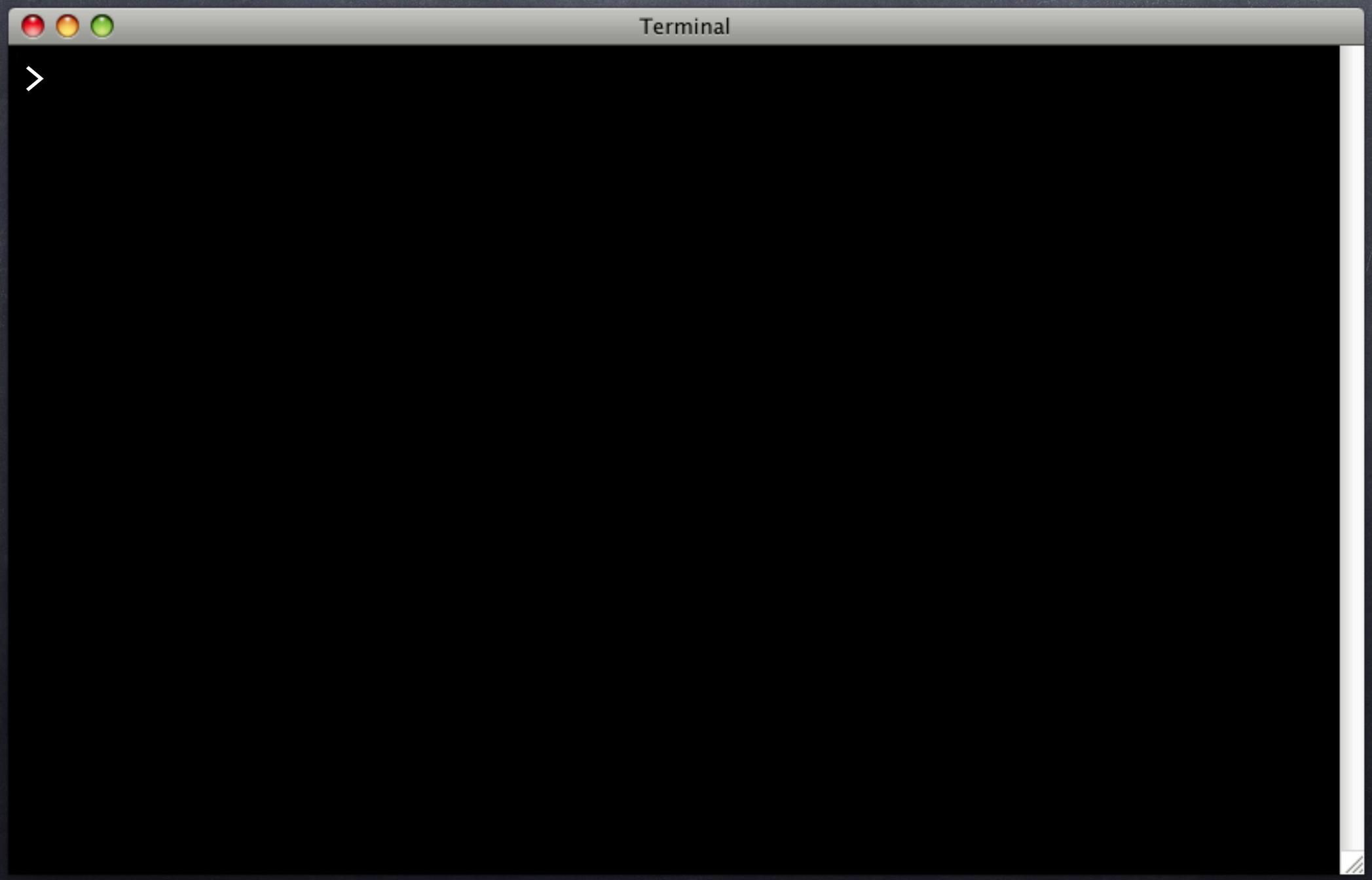
deploy/pgcrypto.sql

Emacs

```
-- Deploy pgcrypto

BEGIN;

CREATE EXTENSION pgcrypto;

COMMIT;
```

--:-- deploy/pgcrypto All (SQL[ansi])

Add pgcrypto



Add pgcrypto

Terminal

```
> sqitch add pgcrypto -n 'Loads pgcrypto extension.'
Adding deploy/pgcrypto.sql
Adding revert/pgcrypto.sql
Adding verify/pgcrypto.sql
Added "pgcrypto" to sqitch.plan
> emacs deploy/pgcrpyto.sql
>
```

Add pgcrypto

Terminal

```
> sqitch add pgcrypto -n 'Loads pgcrypto extension.'
Adding deploy/pgcrypto.sql
Adding revert/pgcrypto.sql
Adding verify/pgcrypto.sql
Added "pgcrypto" to sqitch.plan
> emacs deploy/pgcrpyto.sql
> emacs verify/pgcrpyto.sql
```

verify/pgcrypto.sql

Emacs

```
-- Verify pgcrypto

BEGIN;

-- XXX Add verifications here.

COMMIT;
```

--:-- verify/pgcrypto All (SQL[ansi])

verify/pgcrypto.sql

Emacs

```sql
-- Verify pgcrypto

BEGIN;

SELECT 1/count(*) FROM pg_extension WHERE extname = 'pgcrypto';
SELECT has_function_privilege('crypt(text, text)', 'execute');
SELECT has_function_privilege('gen_salt(text)', 'execute');

COMMIT;
```

--:-- verify/pgcrypto All (SQL[ansi])

verify/pgcrypto.sql

Emacs

```
-- Verify pgcrypto

BEGIN;

SELECT 1/count(*) FROM pg_extension WHERE extname = 'pgcrypto';
SELECT has_function_privilege('crypt(text, text)', 'execute');
SELECT has_function_privilege('gen_salt(text)', 'execute');

COMMIT;
```

--:-- verify/pgcrypto All (SQL[ansi])

verify/pgcrypto.sql

Emacs

```
-- Verify pgcrypto

BEGIN;

SELECT 1/count(*) FROM pg_extension WHERE extname = 'pgcrypto';
SELECT has_function_privilege('crypt(text, text)', 'execute');
SELECT has_function_privilege('gen_salt(text)', 'execute');

COMMIT;
```

--:-- verify/pgcrypto All (SQL[ansi])

How to Modify?

How to Modify?

- Copy insert_user.sql to new deploy file

How to Modify?

- Copy insert_user.sql to new deploy file
- Change that new file

How to Modify?

- Copy insert_user.sql to new deploy file
- Change that new file
- Copy insert_user.sql to new revert file

How to Modify?

- Copy insert_user.sql to new deploy file
- Change that new file
- Copy insert_user.sql to new revert file
- Test it

How to Modify?

- Copy insert_user.sql to new deploy file
- Change that new file
- Copy insert_user.sql to new revert file
- Test it
- Do the same for change_pass.sql

How to Modify?

- Copy insert_user.sql to new deploy file
- Change that new file
- Copy insert_user.sql to new revert file
- Test it
- Do the same for change_pass.sql
- The problem with that...

Terminal

>

Terminal

```
> git diff HEAD^
diff --git a/deploy/insert_user_crypt.sql b/deploy/insert_user_crypto.sql
new file mode 100644
index 0000000..fa8d0c6
--- /dev/null
+++ b/deploy/insert_user_crypt.sql
@@ -0,0 +1,8 @@
+-- requires: users, appuser, pgcrypto
+
+CREATE OR REPLACE FUNCTION insert_user(
+    nickname TEXT,
+    password TEXT
+) RETURNS VOID LANGUAGE SQL AS $$
+    INSERT INTO users values($1, crypt($2, gen_salt('md5')));
+$$;
diff --git a/revert/insert_user_crypt.sql b/revert/insert_user_crypto.sql
new file mode 100644
index 0000000..a7f4e31
--- /dev/null
+++ b/revert/insert_user_crypt.sql
@@ -0,0 +1,8 @@
+-- requires: users, appuser
+
+CREATE OR REPLACE FUNCTION insert_user(
+    nickname TEXT,
+    password TEXT
+) RETURNS VOID LANGUAGE SQL AS $$
+    INSERT INTO users values($1, md5($2));
+$$;
```

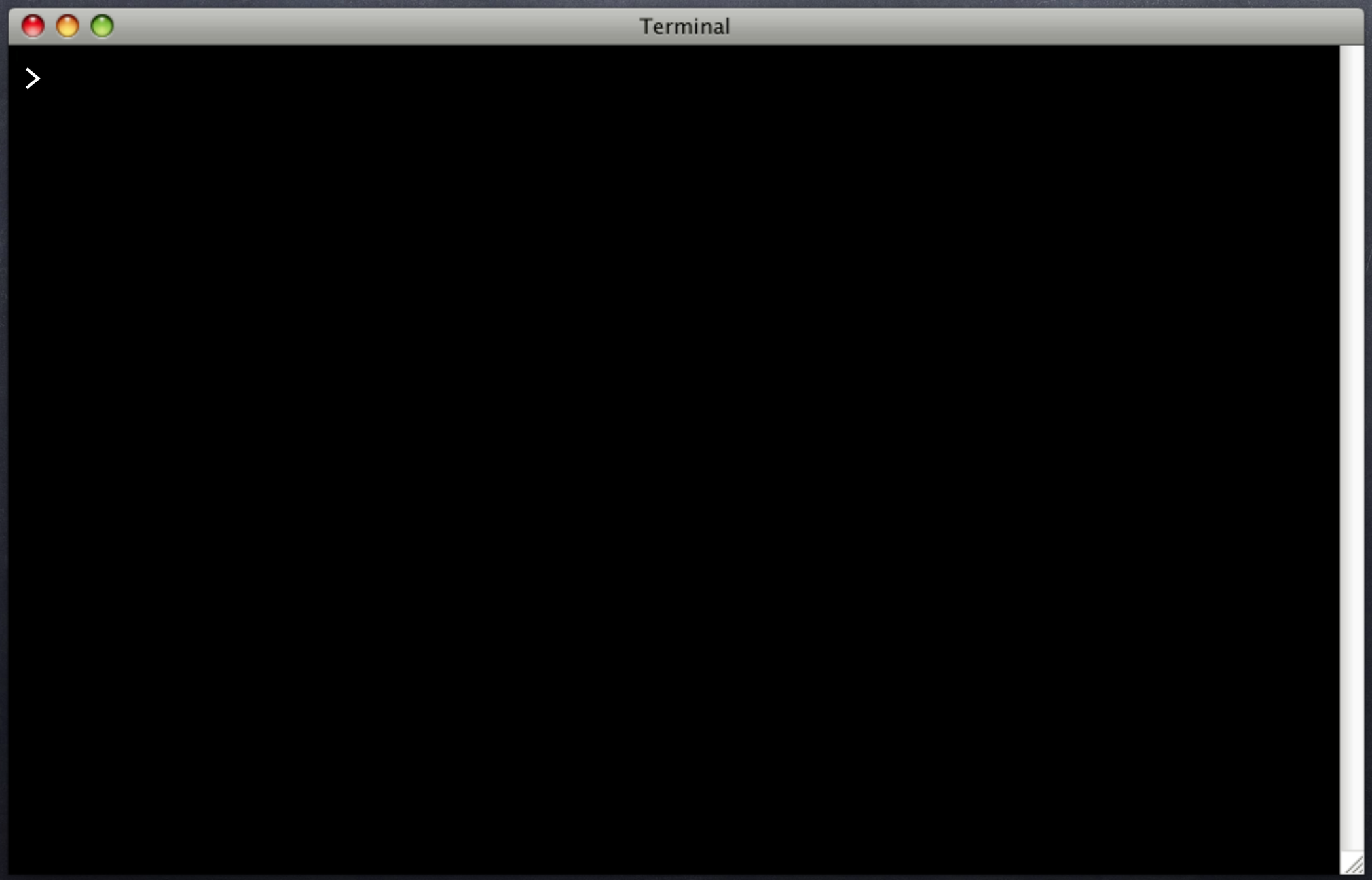
Terminal

```
> git diff HEAD^
diff --git a/deploy/insert_user_crypt.sql b/deploy/insert_user_crypto.sql
new file mode 100644
index 0000000..fa8d0c6
--- /dev/null
+++ b/deploy/insert_user_crypt.sql
@@ -0,0 +1,8 @@
+-- requires: users, appuser, pgcrypto
+
+CREATE OR REPLACE FUNCTION insert_user(
+    nickname TEXT,
+    password TEXT
+) RETURNS VOID LANGUAGE SQL AS $$
+    INSERT INTO users values($1, crypt($2, gen_salt('md5')));
+$$;
diff --git a/revert/insert_user_crypt.sql b/revert/insert_user_crypto.sql
new file mode 100644
index 0000000..a7f4e31
--- /dev/null
+++ b/revert/insert_user_crypt.sql
@@ -0,0 +1,8 @@
+-- requires: users, appuser
+
+CREATE OR REPLACE FUNCTION insert_user(
+    nickname TEXT,
+    password TEXT
+) RETURNS VOID LANGUAGE SQL AS $$
+    INSERT INTO users values($1, md5($2));
+$$;
```

Oy.

Let Sqitch do the work.

Rework It



Rework It

Terminal

```
> sqitch rework insert_user -r pgcrypto \
  -n 'Changes insert_user to use pgcrypto.'
Added "insert_user [insert_user@v1.0.0-dev1 pgcrypto]" to
sqitch.plan.
Modify these files as appropriate:
  * deploy/insert_user.sql
  * revert/insert_user.sql
  * verify/insert_user.sql
>
```

Rework It

Terminal

```
> sqitch rework insert_user -r pgcrypto \
  -n 'Changes insert_user to use pgcrypto.'
Added "insert_user [insert_user@v1.0.0-dev1 pgcrypto]" to
sqitch.plan.
Modify these files as appropriate:
  * deploy/insert_user.sql
  * revert/insert_user.sql
  * verify/insert_user.sql
>
```

Rework It

Terminal

```
> sqitch rework insert_user -r pgcrypto \
  -n 'Changes insert_user to use pgcrypto.'
Added "insert_user [insert_user@v1.0.0-dev1 pgcrypto]" to
sqitch.plan.
Modify these files as appropriate:
  * deploy/insert_user.sql
  * revert/insert_user.sql
  * verify/insert_user.sql
>
```

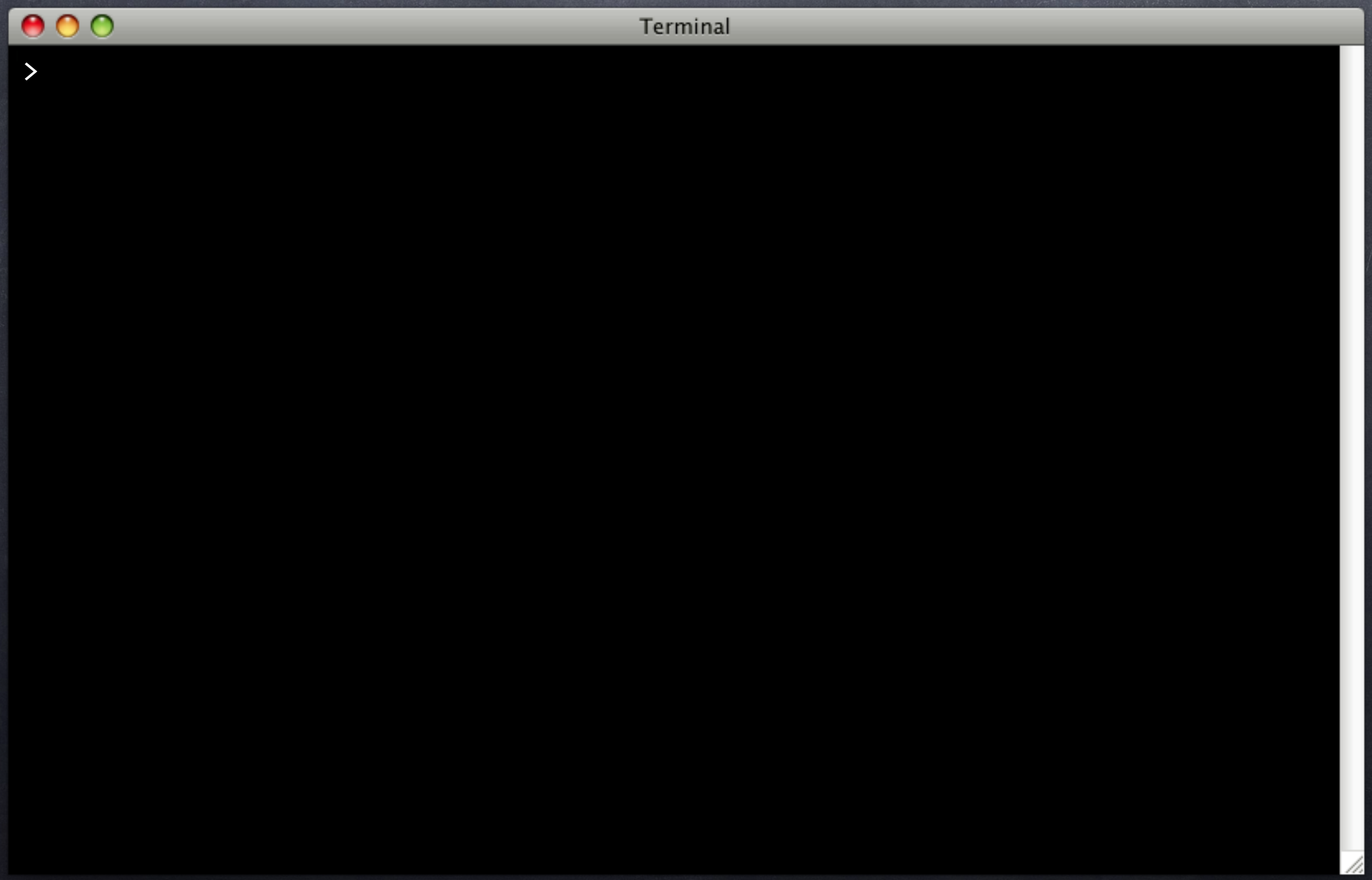
Rework It

Terminal

```
> sqitch rework insert_user -r pgcrypto \
  -n 'Changes insert_user to use pgcrypto.'
Added "insert_user [insert_user@v1.0.0-dev1 pgcrypto]" to
sqitch.plan.
Modify these files as appropriate:
  * deploy/insert_user.sql
  * revert/insert_user.sql
  * verify/insert_user.sql
>
```

Same files?

Same Files?



Same Files?

Terminal

```
> git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committe
#   (use "git checkout -- <file>..." to discard changes in w
#
#   modified:   revert/insert_user.sql
#   modified:   sqitch.plan
#
# Untracked files:
#   (use "git add <file>..." to include in what will be comm
#
#   deploy/insert_user@v1.0.0-dev1.sql
#   revert/insert_user@v1.0.0-dev1.sql
#   verify/insert_user@v1.0.0-dev1.sql
no changes added to commit (use "git add" and/or "git commit
```

Same Files?

Terminal

```
> git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committe
#   (use "git checkout -- <file>..." to discard changes in w
#
#   modified:   revert/insert_user.sql
#   modified:   sqitch.plan
#
# Untracked files:
#   (use "git add <file>..." to include in what will be comm
#
#   deploy/insert_user@v1.0.0-dev1.sql
#   revert/insert_user@v1.0.0-dev1.sql
#   verify/insert_user@v1.0.0-dev1.sql
no changes added to commit (use "git add" and/or "git commit
```

Same Files?

Terminal

```
> git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committe
#   (use "git checkout -- <file>..." to discard changes in w
#
#   modified:   revert/insert_user.sql
#   modified:   sqitch.plan
#
# Untracked files:
#   (use "git add <file>..." to include in what will be comm
#
#   deploy/insert_user@v1.0.0-dev1.sql
#   revert/insert_user@v1.0.0-dev1.sql
#   verify/insert_user@v1.0.0-dev1.sql
no changes added to commit (use "git add" an
```

As of @v1.0.0-dev1

Same Files?

Terminal

```
> git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committe
#   (use "git checkout -- <file>..." to discard changes in w
#
#   modified:   revert/insert_user.sql
#   modified:   sqitch.plan
#
# Untracked files:
#   (use "git add <file>..." to include in what will be comm
#
#   deploy/insert_user@v1.0.0-dev1.sql
#   revert/insert_user@v1.0.0-dev1.sql
#   verify/insert_user@v1.0.0-dev1.sql
no changes added to commit (use "git add" and/or "git commit
```

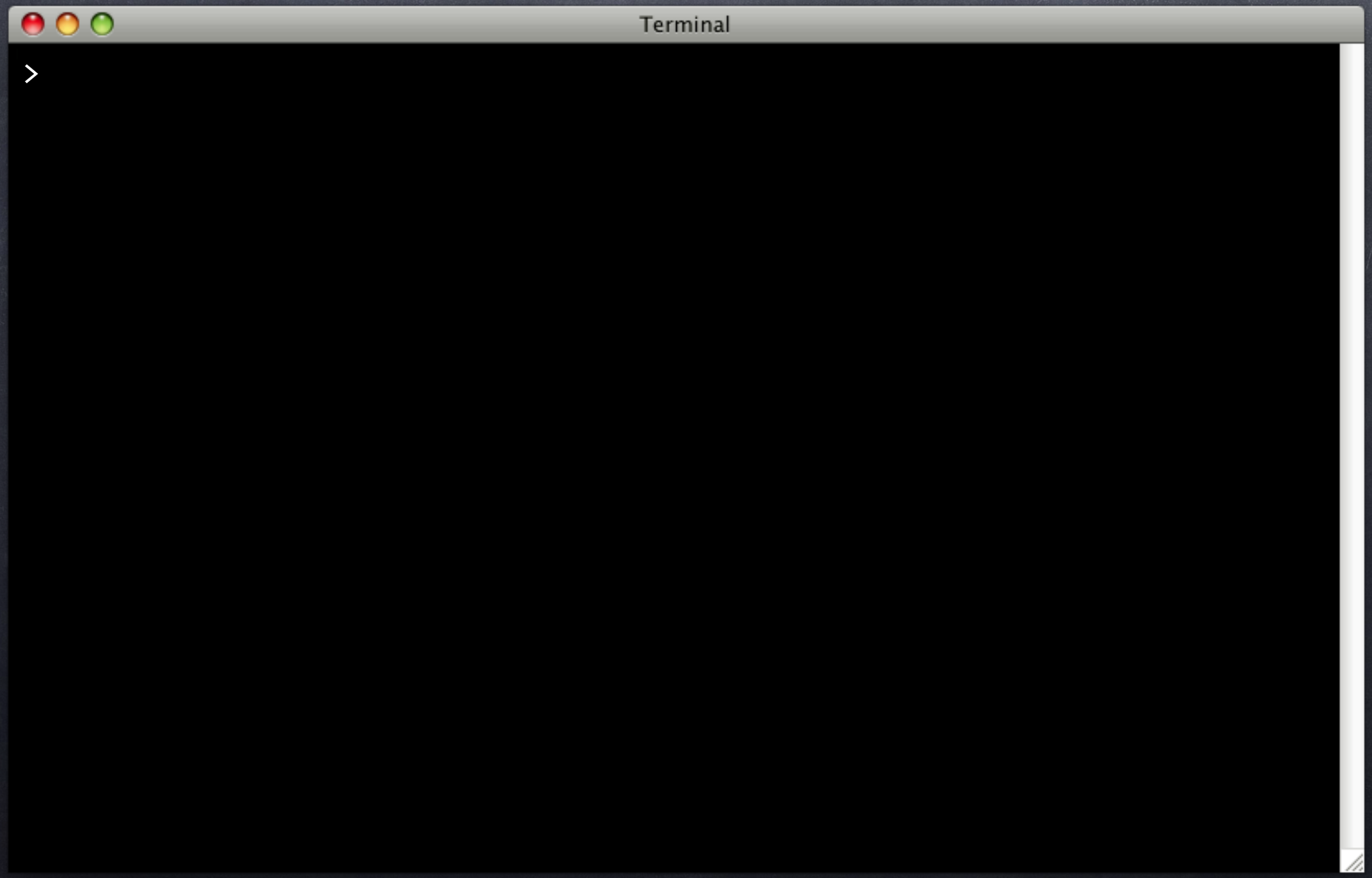
Same Files?

Terminal

```
> git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committe
#   (use "git checkout -- <file>..." to discard changes in w
#
#   modified:   revert/insert_user.sql
#   modified:   sqitch.plan
#
# Untracked files:
#   (use "git add <file>..." to include in
#
#   deploy/insert_user@v1.0.0-dev1.sql
#   revert/insert_user@v1.0.0-dev1.sql
#   verify/insert_user@v1.0.0-dev1.sql
no changes added to commit (use "git add" and/or "git commit
```

Previous deploy becomes revert

What's the Diff?

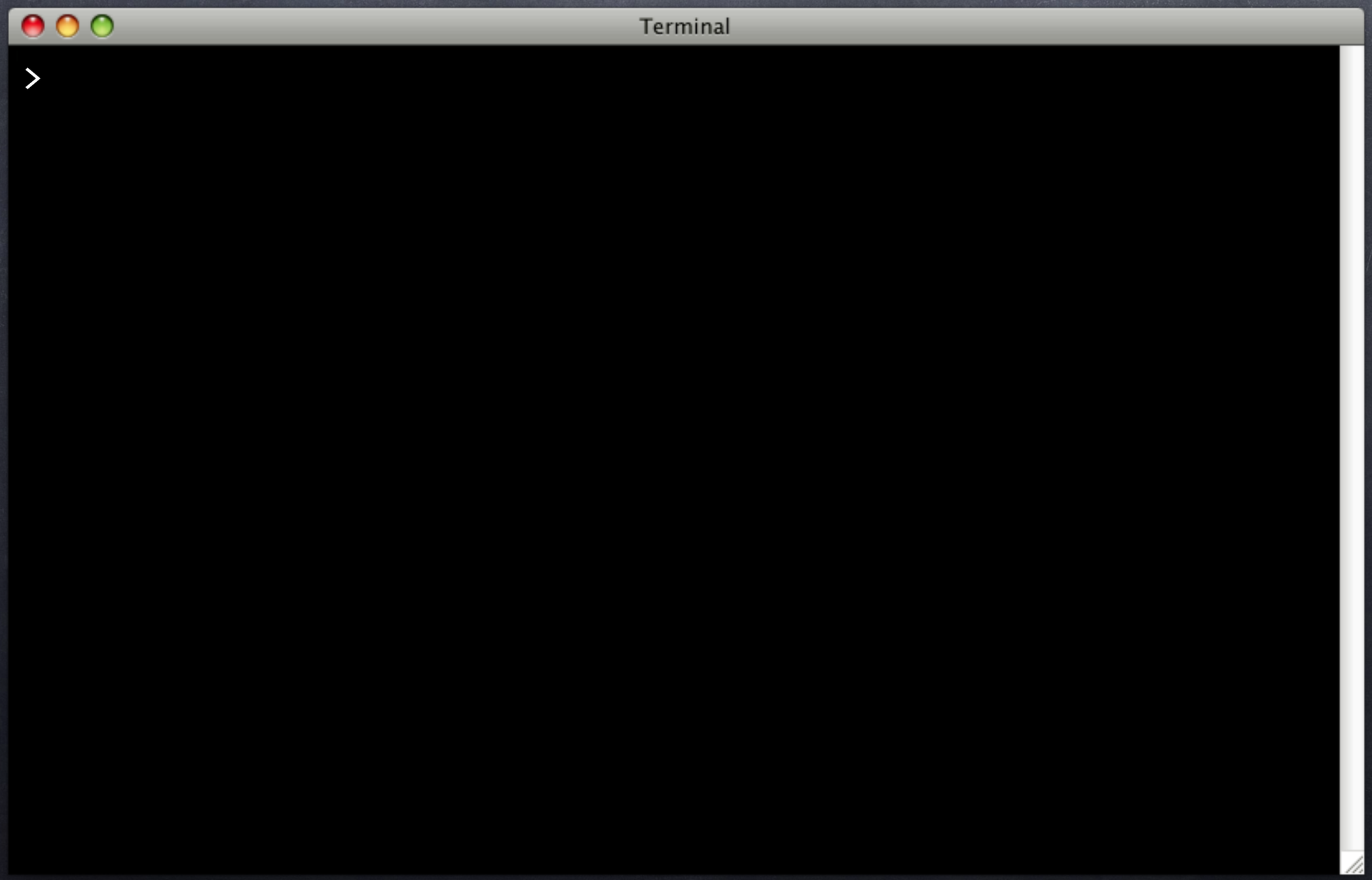


What's the Diff?

Terminal

```
> diff -u deploy/insert_user.sql
@@ -1,6 +1,7 @@
 -- Deploy insert_user
 -- requires: users
 -- requires: appschema
+-- requires: pgcrypto

 BEGIN;

@@ -8,7 +9,7 @@ CREATE OR REPLACE FUNCTION flipr.insert_user(
     nickname TEXT,
     password TEXT
 ) RETURNS VOID LANGUAGE SQL SECURITY DEFINER AS $$
-    INSERT INTO flipr.users VALUES($1, md5($2));
+    INSERT INTO flipr.users values($1, crypt($2, gen_salt('md5')));
 $$;

 COMMIT;
```

Rework change_pass



Rework change_pass

Terminal

```
> sqitch rework change_pass -r pgcrypto \
  -n 'Change change_pass to use pgcrypto.'
Added "change_pass [change_pass@v1.0.0-dev1 pgcrypto]" to
sqitch.plan.
Modify these files as appropriate:
  * deploy/change_pass.sql
  * revert/change_pass.sql
  * verify/change_pass.sql
>
```

Rework change_pass

Terminal

```
> sqitch rework change_pass -r pgcrypto  \
  -n 'Change change_pass to use pgcrypto.'
Added "change_pass [change_pass@v1.0.0-dev1 pgcrypto]" to
sqitch.plan.
Modify these files as appropriate:
  * deploy/change_pass.sql
  * revert/change_pass.sql
  * verify/change_pass.sql
>
```

Rework change_pass

Terminal

```
> sqitch rework change_pass -r pgcrypto  \
  -n 'Change change_pass to use pgcrypto.'
Added "change_pass [change_pass@v1.0.0-dev1 pgcrypto]" to
sqitch.plan.
Modify these files as appropriate:
  * deploy/change_pass.sql
  * revert/change_pass.sql
  * verify/change_pass.sql
>
```

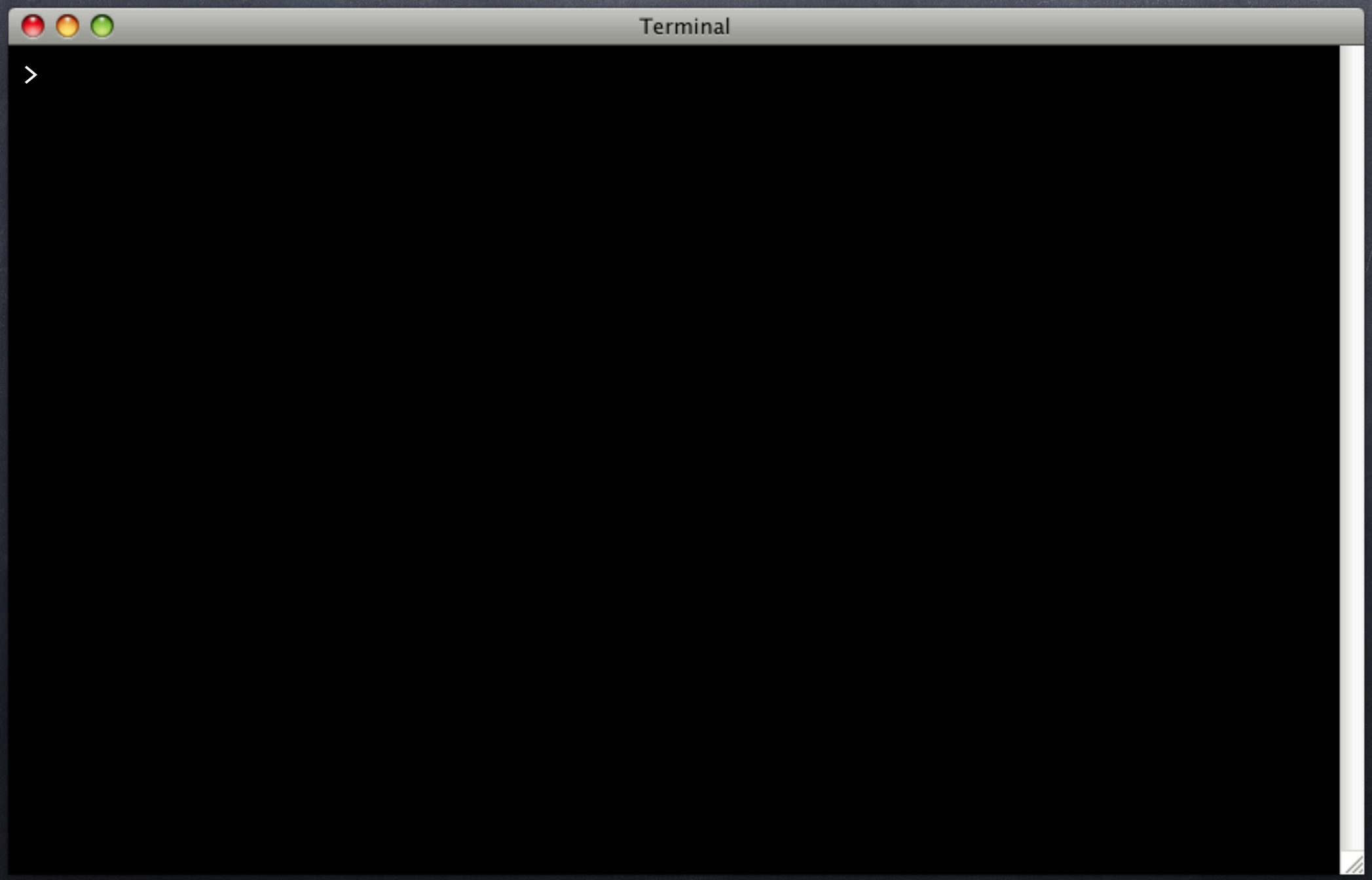
Rework change_pass

Terminal

```
> sqitch rework change_pass -r pgcrypto  \
  -n 'Change change_pass to use pgcrypto.'
Added "change_pass [change_pass@v1.0.0-dev1 pgcrypto]" to
sqitch.plan.
Modify these files as appropriate:
  * deploy/change_pass.sql
  * revert/change_pass.sql
  * verify/change_pass.sql
>
```

Same files?

What's the Diff?

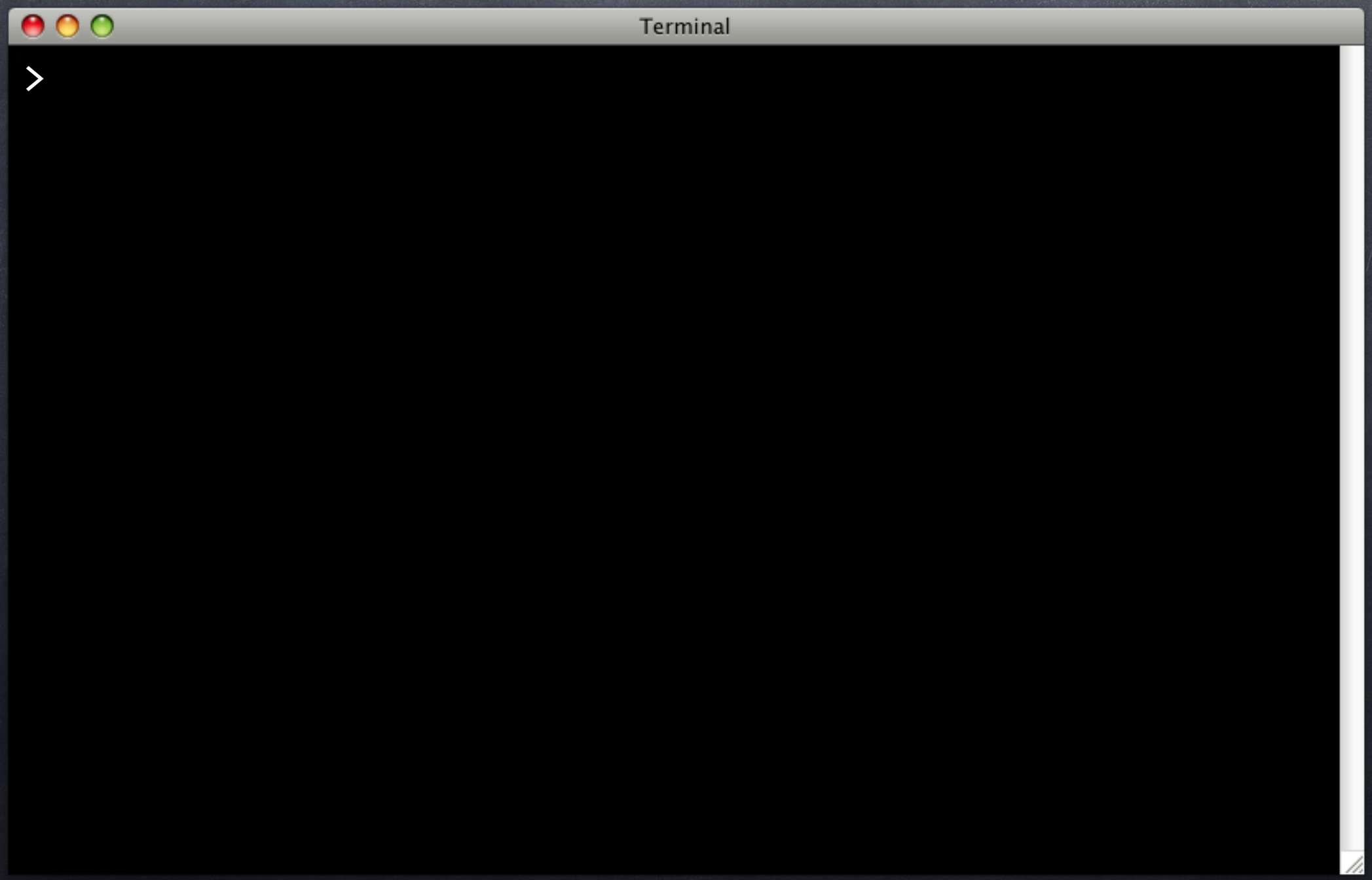


What's the Diff?

Terminal

```
> diff -u deploy/change_pass.sql
@@ -1,6 +1,7 @@
 -- Deploy change_pass
 -- requires: users
 -- requires: appschema
+-- requires: pgcrypto

 BEGIN;

@@ -11,9 +12,9 @@ CREATE OR REPLACE FUNCTION flipr.change_pass(
 ) RETURNS BOOLEAN LANGUAGE plpgsql SECURITY DEFINER AS $$
 BEGIN
     UPDATE flipr.users
-       SET password = md5($3)
+       SET password = crypt($3, gen_salt('md5'))
      WHERE nickname = $1
-       AND password = md5($2);
+       AND password = crypt($2, password);
     RETURN FOUND;
 END;
 $$;
```

Send it Up!



Send it Up!

Terminal

```
> sqitch deploy
Deploying changes to flipr_test
  + insert_user .. ok
  + change_pass .. ok
>
```

Send it Up!

Terminal

```
> sqitch deploy
Deploying changes to flipr_test
  + insert_user .. ok
  + change_pass .. ok
> psql -d flipr_test -c "
    DELETE FROM users;
    SELECT insert_user('foo', 'secr3t'),
           insert_user('bar', 'secr3t');
    SELECT nickname, password FROM users;
"
 nickname |                  password
----------+--------------------------------
 foo      | $1$l6OEKyF3$kv5ae7505ROub75d9QKTh/
 bar      | $1$J4NJDgaJ$578i9Lt6b8ohJwi6WhNNO1
>
```

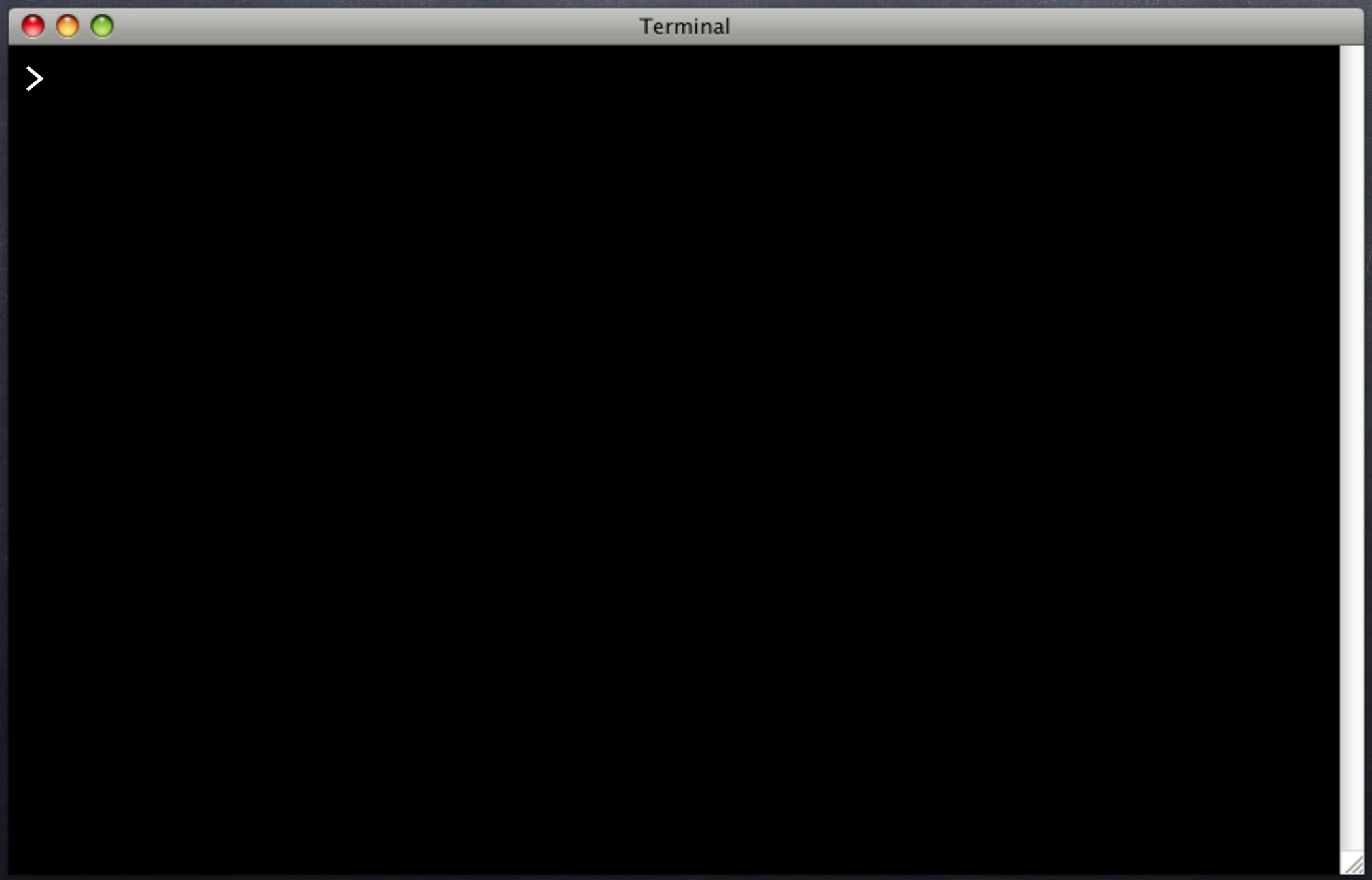
Send it Up!

Terminal

```
> sqitch deploy
Deploying changes to flipr_test
  + insert_user .. ok
  + change_pass .. ok
> psql -d flipr_test -c "
    DELETE FROM users;
    SELECT insert_user('foo', 'secr3t'),
           insert_user('bar', 'secr3t');
    SELECT nickname, password FROM users;
"
 nickname |                 password
----------+------------------------------------
 foo      | $1$l6OEKyF3$kv5ae7505ROub75d9QKTh/
 bar      | $1$J4NJDgaJ$578i9Lt6b8ohJwi6WhNNO1
>
```

\o/

Can We Go Back?



Can We Go Back?

Terminal

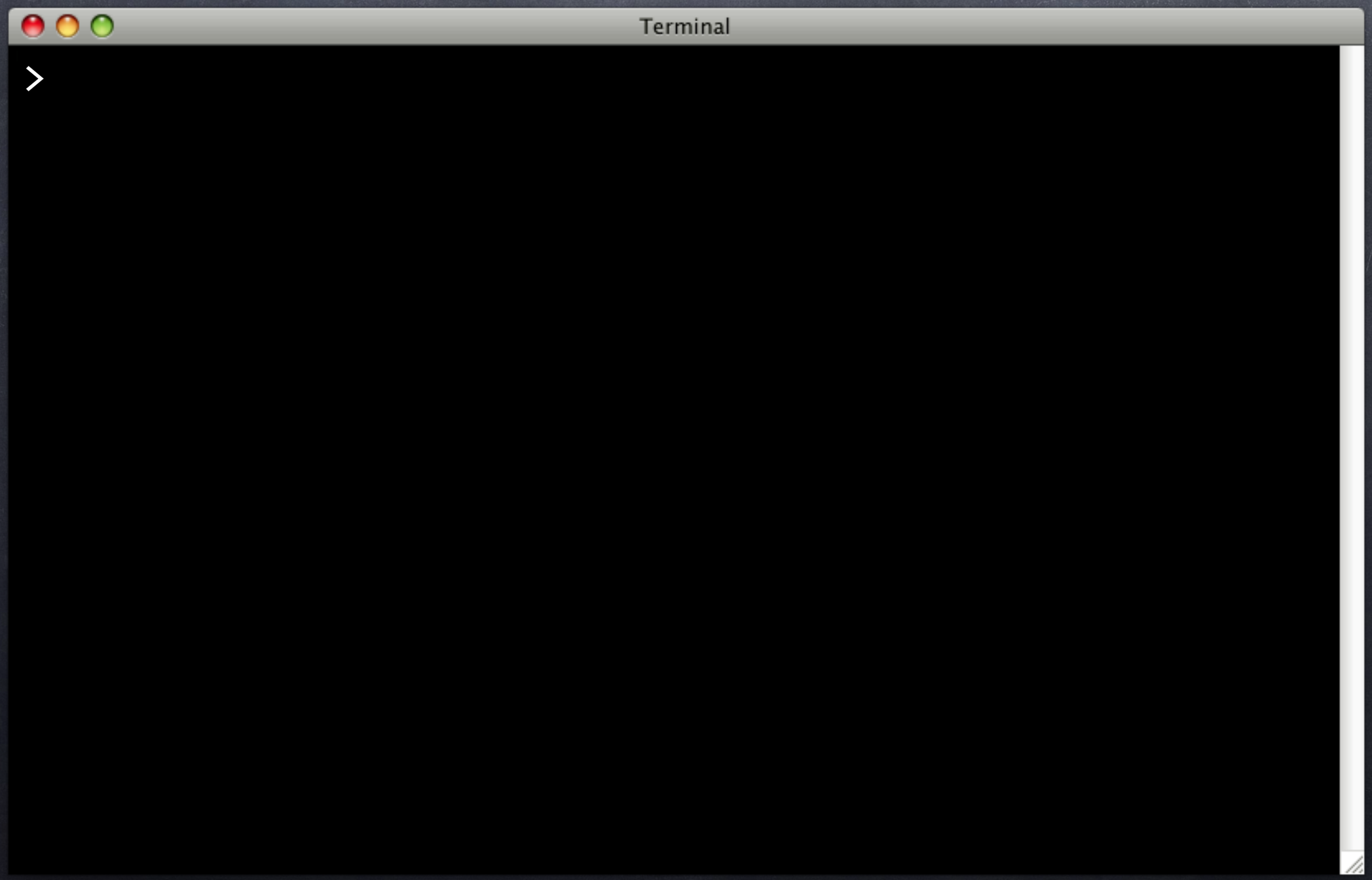
```
> sqitch revert --to @HEAD^^
Reverting changes to pgcrypto from flipr_test
  - change_pass .. ok
  - insert_user .. ok
>
```

Can We Go Back?

Terminal

```
> sqitch revert --to @HEAD^^
Reverting changes to pgcrypto from flipr_test
  - change_pass .. ok
  - insert_user .. ok
> psql -d flipr_test -c "
    DELETE FROM users;
    SELECT insert_user('foo', 'secr3t'),
           insert_user('bar', 'secr3t');
    SELECT nickname, password FROM users;
"
 nickname |              password
----------+----------------------------------
 foo      | 9695da4dd567a19f9b92065f240c6725
 bar      | 9695da4dd567a19f9b92065f240c6725
>
```

Verify How?



Verify How?

Terminal

```
> emacs verify/insert_user.sql
```

verify/insert_user.sql

Emacs

```
-- Verify insert_user

BEGIN;

SELECT has_function_privilege(
    'flipr.insert_user(text, text)',
    'execute'
);

COMMIT;
```

--:-- verify/insert_u All (SQL[ansi])

verify/insert_user.sql

Emacs

```sql
-- Verify insert_user

BEGIN;

SELECT has_function_privilege(
    'flipr.insert_user(text, text)',
    'execute'
);

SELECT 1/COUNT(*)
  FROM pg_catalog.pg_proc
 WHERE proname = 'insert_user'
   AND pg_get_functiondef(oid)
       LIKE $$%crypt($2, gen_salt('md5'))%$$;

COMMIT;
```

--:-- verify/insert_u All (SQL[ansi])

verify/insert_user.sql

Emacs

```
-- Verify insert_user

BEGIN;

SELECT has_function_privilege(
    'flipr.insert_user(text, text)',
    'execute'
);

SELECT 1/COUNT(*)
  FROM pg_catalog.pg_proc
 WHERE proname = 'insert_user'
   AND pg_get_functiondef(oid)
       LIKE $$%crypt($2, gen_salt('md5'))%$$;

COMMIT;
```

--:-- verify/insert_u All (SQL[ansi])

Yeah, compare source.

Verify How?

Terminal

```
> emacs verify/insert_user.sql
>
```

Verify How?

Terminal

```
> emacs verify/insert_user.sql
> emacs verify/change_pass.sql
```

verify/change_pass.sql

Emacs

```
-- Verify change_pass

BEGIN;

SELECT has_function_privilege(
    'flipr.change_pass(text, text, text)',
    'execute'
);

COMMIT;
```

--:-- verify/change_p All (SQL[ansi])

verify/change_pass.sql

Emacs

```sql
-- Verify change_pass

BEGIN;

SELECT has_function_privilege(
    'flipr.change_pass(text, text, text)',
    'execute'
);

SELECT 1/COUNT(*)
  FROM pg_catalog.pg_proc
 WHERE proname = 'change_pass'
   AND pg_get_functiondef(oid)
       LIKE $$%crypt($3, gen_salt('md5'))%$$;

COMMIT;
```

--:-- verify/change_p All (SQL[ansi])

Let's Go!

Terminal

```
> emacs verify/insert_user.sql
> emacs verify/change_pass.sql
>
```

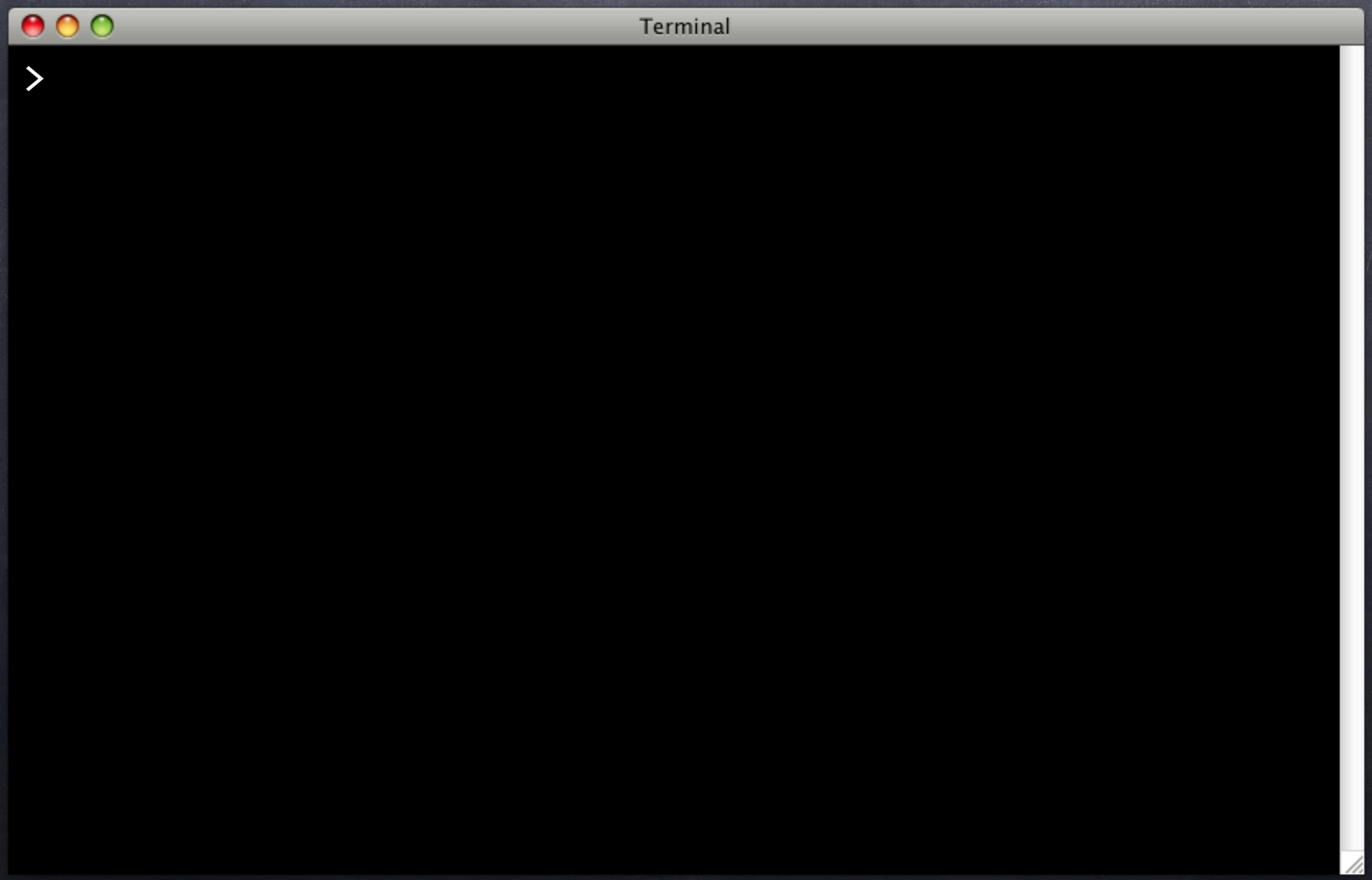
Let's Go!

Terminal

```
> emacs verify/insert_user.sql
> emacs verify/change_pass.sql
> sqitch deploy
Deploying changes to flipr_test
  + insert_user .. ok
  + change_pass .. ok
>
```

Shazam!

What About Bundling?



What About Bundling?

Terminal

```
> sqitch tag v1.0.0-b1 -n 'Tag v1.0.0-b1.'
> git tag v1.0.0-b1 -am 'Tag v1.0.0-b1'
> sqitch bundle
Bundling into bundle
Writing config
Writing plan
Writing scripts
  + appschema
  + users
  + insert_user
  + change_pass @v1.0.0-dev1
  + pgcrypto
  + insert_user
  + change_pass @v1.0.0-b1
>
```

What About Bundling?

Terminal

```
> sqitch tag v1.0.0-b1 -n 'Tag v1.0.0-b1.'
> git tag v1.0.0-b1 -am 'Tag v1.0.0-b1'
> sqitch bundle
Bundling into bundle
Writing config
Writing plan
Writing scripts
  + appschema
  + users
  + insert_user
  + change_pass @v1.0.0-dev1
  + pgcrypto
  + insert_user
  + change_pass @v1.0.0-b1
>
```

What About Bundling?

Terminal

```
> sqitch tag v1.0.0-b1 -n 'Tag v1.0.0-b1.'
> git tag v1.0.0-b1 -am 'Tag v1.0.0-b1'
> sqitch bundle
Bundling into bundle
Writing config
Writing plan
Writing scripts
  + appschema
  + users
  + insert_user
  + change_pass @v1.0.0-dev1
  + pgcrypto
  + insert_user
  + change_pass @v1.0.0-b1
>
```

What About Bundling?

Terminal

```
> sqitch tag v1.0.0-b1 -n 'Tag v1.0.0-b1.'
> git tag v1.0.0-b1 -am 'Tag v1.0.0-b1'
> sqitch bundle
Bundling into bundle
Writing config
Writing plan
Writing scripts
  + appschema
  + users
  + insert_user
  + change_pass @v1.0.0-dev1
  + pgcrypto
  + insert_user
  + change_pass @v1.0.0-b1
> emacs bundle/sqitch.plan
```

What's the Plan?

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
@v1.0.0-dev1 2013-01-03T21:33:00Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2013-01-03T21:36:22Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2013-01-03T22:09:50Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2013-01-03T22:13:58Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2013-01-03T22:25:42Z Marge N. O'Vera <marge@example.com> # Tag
--:-- bundle/sqitch.p  All  (SQL[ansi])----------
```

What's the Plan?

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
@v1.0.0-dev1 2013-01-03T21:33:00Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2013-01-03T21:36:22Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2013-01-03T22:09:50Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2013-01-03T22:13:58Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2013-01-03T22:25:42Z Marge N. O'Vera <marge@example.com> # Tag
--:-- bundle/sqitch.p   All   (SQL[ansi])----------
```

What's the Plan?

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
@v1.0.0-dev1 2013-01-03T21:33:00Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2013-01-03T21:36:22Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2013-01-03T22:09:50Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2013-01-03T22:13:58Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2013-01-03T22:25:42Z Marge N. O'Vera <marge@example.com> # Tag
--:-- bundle/sqitch.p   All   (SQL[ansi])----------
```

What's the Plan?

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
@v1.0.0-dev1 2013-01-03T21:33:00Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2013-01-03T21:36:22Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2013-01-03T22:09:50Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2013-01-03T22:13:58Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2013-01-03T22:25:42Z Marge N. O'Vera <marge@example.com> # Tag
--:-- bundle/sqitch.p   All   (SQL[ansi])----------
```

What's the Plan?

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
@v1.0.0-dev1 2013-01-03T21:33:00Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2013-01-03T21:36:22Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2013-01-03T22:09:50Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2013-01-03T22:13:58Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2013-01-03T22:25:42Z Marge N. O'Vera <marge@example.com> # Tag
--:-- bundle/sqitch.p   All   (SQL[ansi])----------
```

What's the Plan?

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
@v1.0.0-dev1 2013-01-03T21:33:00Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2013-01-03T21:36:22Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2013-01-03T22:09:50Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2013-01-03T22:13:58Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2013-01-03T22:25:42Z Marge N. O'Vera <marge@example.com> # Tag
--:-- bundle/sqitch.p  All  (SQL[ansi])----------
```

What's the Plan?

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
@v1.0.0-dev1 2013-01-03T21:33:00Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2013-01-03T21:36:22Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2013-01-03T22:09:50Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2013-01-03T22:13:58Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2013-01-03T22:25:42Z Marge N. O'Vera <marge@example.com> # Tag
--:-- bundle/sqitch.p   All   (SQL[ansi])----------
```

What's the Plan?

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
@v1.0.0-dev1 2013-01-03T21:33:00Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2013-01-03T21:36:22Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2013-01-03T22:09:50Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2013-01-03T22:13:58Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2013-01-03T22:25:42Z Marge N. O'Vera <marge@example.com> # Tag
--:-- bundle/sqitch.p   All   (SQL[ansi])----------
```

What's the Plan?

Emacs

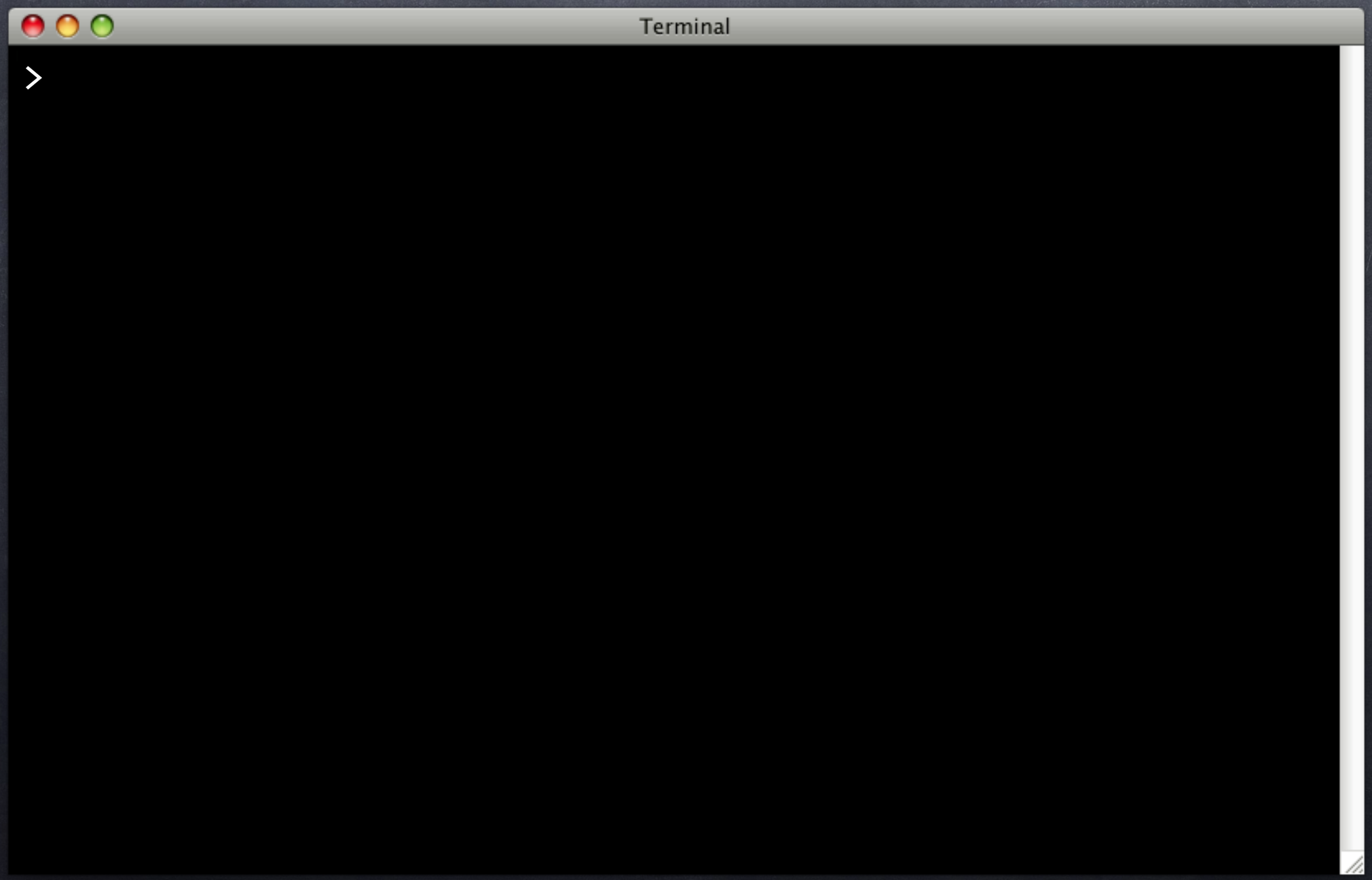
```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
@v1.0.0-dev1 2013-01-03T21:33:00Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2013-01-03T21:36:22Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2013-01-03T22:09:50Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2013-01-03T22:13:58Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2013-01-03T22:25:42Z Marge N. O'Vera <marge@example.com> # Tag
--:-- bundle/sqitch.p   All   (SQL[ansi])----------
```

What's the Plan?

Emacs

```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-12-29T01:08:29Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-12-31T19:38:06Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-12-31T22:44:27Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-12-31T22:44:38Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
@v1.0.0-dev1 2013-01-03T21:33:00Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2013-01-03T21:36:22Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2013-01-03T22:09:50Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2013-01-03T22:13:58Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2013-01-03T22:25:42Z Marge N. O'Vera <marge@example.com> # Tag
--:-- bundle/sqitch.p   All   (SQL[ansi])----------
```

Make it So



Make it So

Terminal

```
> cd bundle
>
```

Make it So

Terminal

```
> cd bundle
> sqitch --db-name flipr_stage deploy
Deploying changes to flipr_stage
  + pgcrypto ................... ok
  + insert_user ............... ok
  + change_pass @v1.0.0-b1 .. ok
>
```

Make it So

Terminal

```
> cd bundle
> sqitch --db-name flipr_stage deploy
Deploying changes to flipr_stage
  + pgcrypto ................. ok
  + insert_user .............. ok
  + change_pass @v1.0.0-b1 .. ok
> sqitch --db-name flipr_stage status
# On database flipr_stage
# Project:  flipr
# Change:   6238d88bb11fa0e9ac9b1a90831bb45f16a8e19a
# Name:     change_pass
# Tag:      @v1.0.0-b1
# Deployed: 2013-01-08 14:02:42 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
```

Make it So

Terminal

```
> cd bundle
> sqitch --db-name flipr_stage deploy
Deploying changes to flipr_stage
  + pgcrypto ................. ok
  + insert_user .............. ok
  + change_pass @v1.0.0-b1 .. ok
> sqitch --db-name flipr_stage status
# On database flipr_stage
# Project:  flipr
# Change:   6238d88bb11fa0e9ac9b1a90831bb45f16a8e19a
# Name:     change_pass
# Tag:      @v1.0.0-b1
# Deployed: 2013-01-08 14:02:42 -0800
# By:       Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
```

Ship It!

Other Commands

Other Commands

- help - Get it

Other Commands

- help - Get it
- log - Like git log

Other Commands

- help - Get it
- log - Like git log
- rebase - Revert and deploy in one command

Other Commands

- help - Get it
- log - Like git log
- rebase - Revert and deploy in one command
- check - Validate plan

Forthcoming

Forthcoming

- Port to SQLite

Forthcoming

- Port to SQLite
- Add VCS/SCM integration

Forthcoming

- Port to SQLite
- Add VCS/SCM integration
  - Sync tags

Forthcoming

- Port to SQLite
- Add VCS/SCM integration
  - Sync tags
  - Fetch reworked files from history

Forthcoming

- Port to SQLite
- Add VCS/SCM integration
  - Sync tags
  - Fetch reworked files from history
  - Bootstrap plan from history

Forthcoming

Forthcoming

- Add plan modification commands:

Forthcoming

- Add plan modification commands:
  - Delete change

Forthcoming

- Add plan modification commands:
  - Delete change
  - Delete tag

Forthcoming

- Add plan modification commands:
  - Delete change
  - Delete tag
  - Reorder changes

Forthcoming

- Add plan modification commands:
  - Delete change
  - Delete tag
  - Reorder changes
  - Modify dependencies

Fork It

Fork It

- http://sqitch.org/

Fork It

- http://sqitch.org/
- https://github.com/theory/sqitch/

Fork It

- http://sqitch.org/
- https://github.com/theory/sqitch/
- Opinions wanted

Fork It

- http://sqitch.org/
- https://github.com/theory/sqitch/
- Opinions wanted
- Coders wanted

Fork It

- http://sqitch.org/
- https://github.com/theory/sqitch/
- Opinions wanted
- Coders wanted
- Doc writers and web designers wanted!

Fork It

- http://sqitch.org/
- https://github.com/theory/sqitch/
- Opinions wanted
- Coders wanted
- Doc writers and web designers wanted!
- Make it great!

Thank you.

Sane SQL Change Management with Sqitch

David E. Wheeler
http://sqitch.org/

PDX.pm
January 9, 2013

Text: Attribution-Noncommercial-Share Alike 3.0 United States:
http://creativecommons.org/licenses/by-nc-sa/3.0/us/
Images licensed independently and © Their respective owners.

iovation