

Sane SQL Change Management with Sqitch

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

PDXPUG
September 20, 2012



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.



Whats Wrong with Migrations?

Whats Wrong with Migrations?

- ➊ Incomplete mini-language

Whats Wrong with Migrations?

- ➊ Incomplete mini-language
- ➋ 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

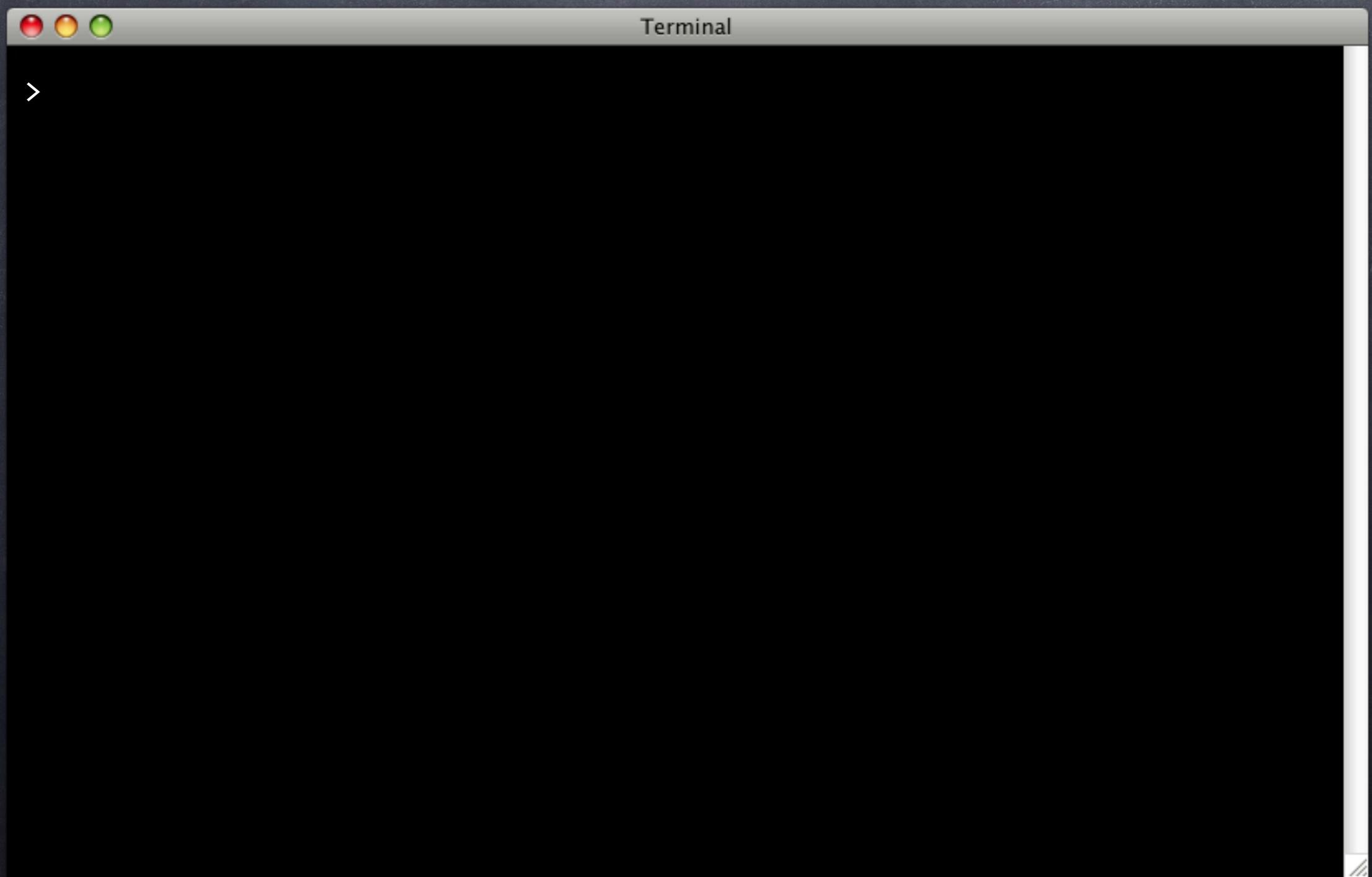
What about SQL Migrations?

- ~~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

- ➊ Paste entire function to new “up” script

Not So Much

- ➊ Paste entire function to new “up” script
- ➋ 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

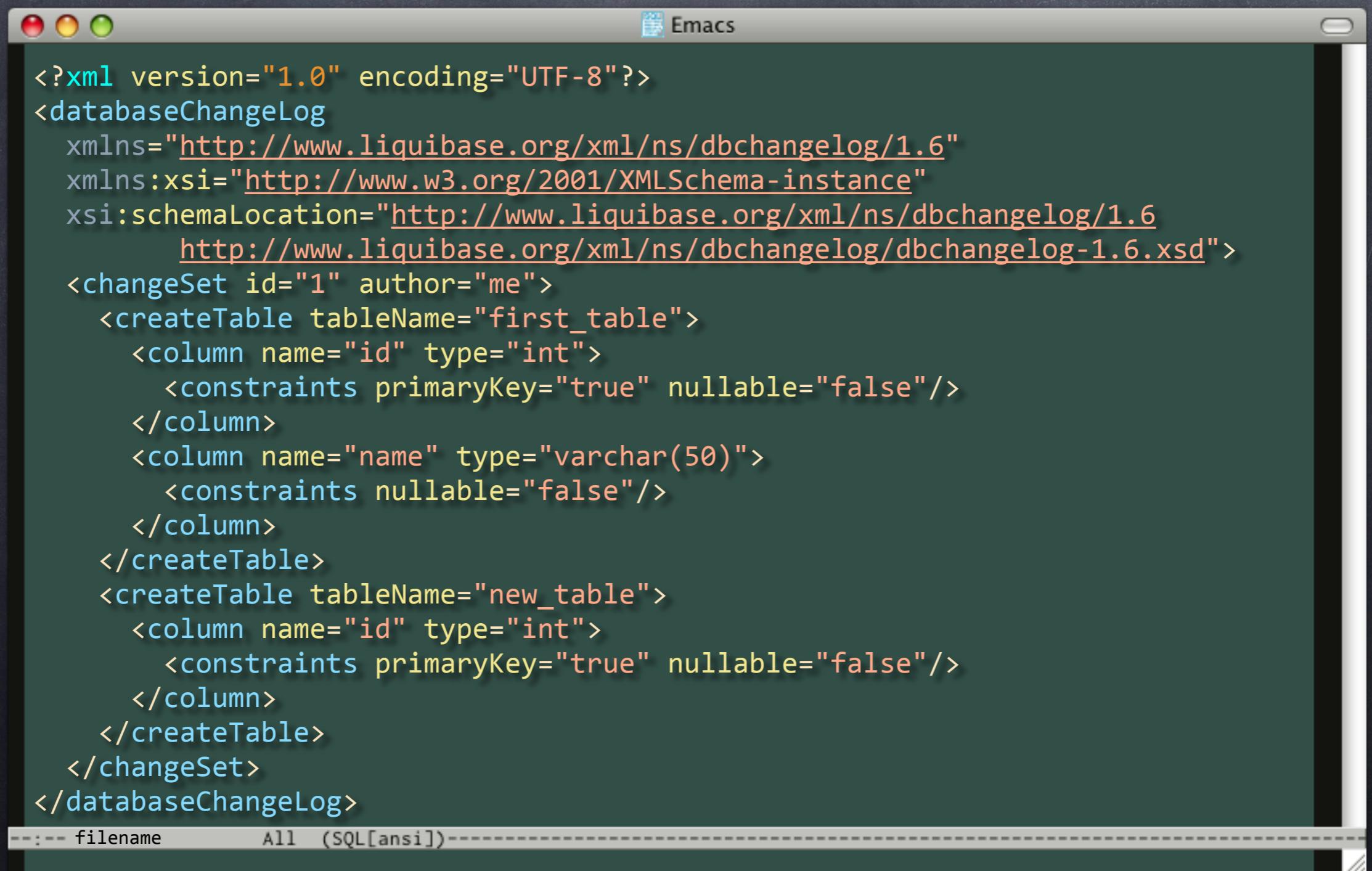
Not So Much

- ⦿ 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

What about Liquibase?



What about Liquibase?

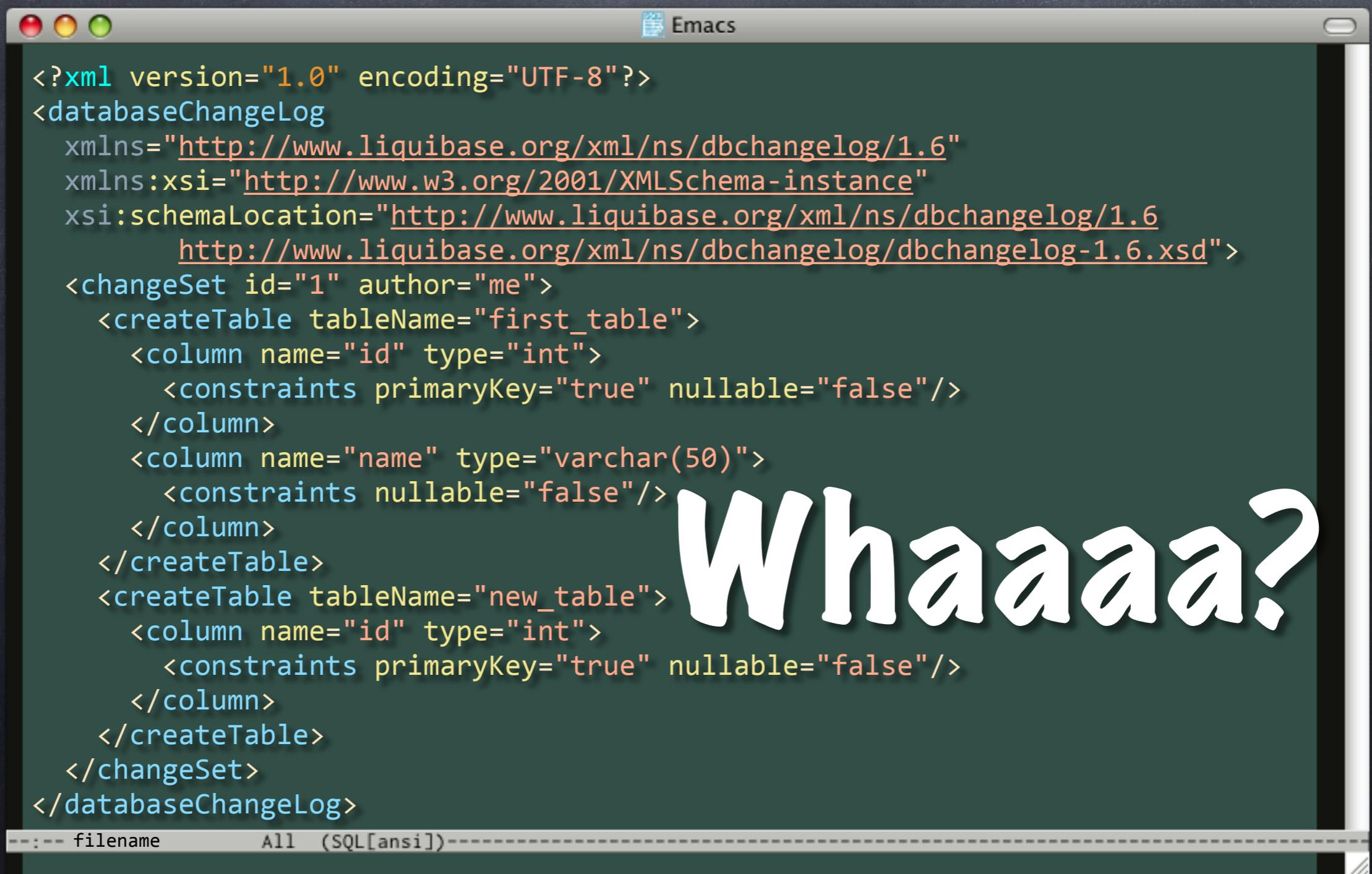


The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". The main buffer contains XML code for a database change log, specifically version 1.6 of the Liquibase schema. The code defines two tables: "first_table" and "new_table", each with two columns: "id" (primary key, int type) and "name" (varchar(50) type). The XML uses color-coded syntax highlighting for tags and attributes.

```
<?xml version="1.0" encoding="UTF-8"?>
<databaseChangeLog
    xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.6"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.6
        http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.6.xsd">
    <changeSet id="1" author="me">
        <createTable tableName="first_table">
            <column name="id" type="int">
                <constraints primaryKey="true" nullable="false"/>
            </column>
            <column name="name" type="varchar(50)">
                <constraints nullable="false"/>
            </column>
        </createTable>
        <createTable tableName="new_table">
            <column name="id" type="int">
                <constraints primaryKey="true" nullable="false"/>
            </column>
        </createTable>
    </changeSet>
</databaseChangeLog>
```

-- filename All (SQL[ansi]) --

What about Liquibase?



```
<?xml version="1.0" encoding="UTF-8"?>
<databaseChangeLog
    xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.6"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.6
        http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.6.xsd">
    <changeSet id="1" author="me">
        <createTable tableName="first_table">
            <column name="id" type="int">
                <constraints primaryKey="true" nullable="false"/>
            </column>
            <column name="name" type="varchar(50)">
                <constraints nullable="false"/>
            </column>
        </createTable>
        <createTable tableName="new_table">
            <column name="id" type="int">
                <constraints primaryKey="true" nullable="false"/>
            </column>
        </createTable>
    </changeSet>
</databaseChangeLog>
```

----- filename All (SQL[ansi])-----

Whaaaa?

What about Liquibase?



Emacs

```
<?xml version="1.0" encoding="UTF-8"?>
<databaseChangeLog
    xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.6"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.6
        http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.6.xsd">
    <changeSet id="1" author="me">
        <createTable tableName="first_table">
            <column name="id" type="int">
                <constraints primaryKey="true" nullable="false"/>
            </column>
            <column name="name" type="varchar(50)">
                <constraints nullable="false"/>
            </column>
        </createTable>
        <createTable tableName="new_table">
            <column name="id" type="int">
                <constraints primaryKey="true" nullable="false"/>
            </column>
        </createTable>
    </changeSet>
</databaseChangeLog>
```

--- filename All (SQL[ansi])---

Whaaaa?
GTFO!

depesz's Versioning?

depesz's Versioning?

- <https://github.com/depesz/versioning>

depesz's Versioning?

- <https://github.com/depesz/versioning>
- Nice dependency specification

depesz's Versioning?

- <https://github.com/depesz/versioning>
- Nice dependency specification
- Tight PostgreSQL integration

depesz's Versioning?

- <https://github.com/depesz/versioning>
- Nice dependency specification
- Tight PostgreSQL integration
- No VCS integration

depesz's Versioning?

- <https://github.com/depesz/versioning>
- Nice dependency specification
- Tight PostgreSQL integration
- No VCS integration
- No tools

depesz's Versioning?

- <https://github.com/depesz/versioning>
- Nice dependency specification
- Tight PostgreSQL integration
- No VCS integration
- No tools
- Managing procedures still a PITA

Introducing Sqitch

Sq—what?

sql changes

Sq—what?

sq ch

Sq—what?

sqitch

Sq—what?

sqitch

There is no “u”

Sqitch Philosophy

Sqitch Philosophy

- ➊ No opinions

Sqitch Philosophy

- ➊ No opinions
- ➋ Native scripting

Sqitch Philosophy

- ⦿ No opinions
- ⦿ Native scripting
- ⦿ Dependency resolution

Sqitch Philosophy

- ⦿ No opinions
- ⦿ Native scripting
- ⦿ Dependency resolution
- ⦿ No numbering

Sqitch Philosophy

- ⦿ No opinions
- ⦿ Native scripting
- ⦿ Dependency resolution
- ⦿ No numbering
- ⦿ Distribution bundling

Sqitch Philosophy

- ⦿ No opinions
- ⦿ Native scripting
- ⦿ Dependency resolution
- ⦿ No numbering
- ⦿ Distribution bundling
- ⦿ VCS integration



Sqitch Philosophy

Sqitch Philosophy

- ➊ Reduced duplication

Sqitch Philosophy

- ➊ Reduced duplication
- ➋ Built-in configuration

Sqitch Philosophy

- ⦿ Reduced duplication
- ⦿ Built-in configuration
- ⦿ Deployment planning

Sqitch Philosophy

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

Sqitch Philosophy

- Reduced duplication
- Built-in configuration
- 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
- add
- tag
- state
- plan

Sqitch Terminology

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

Sqitch Terminology

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

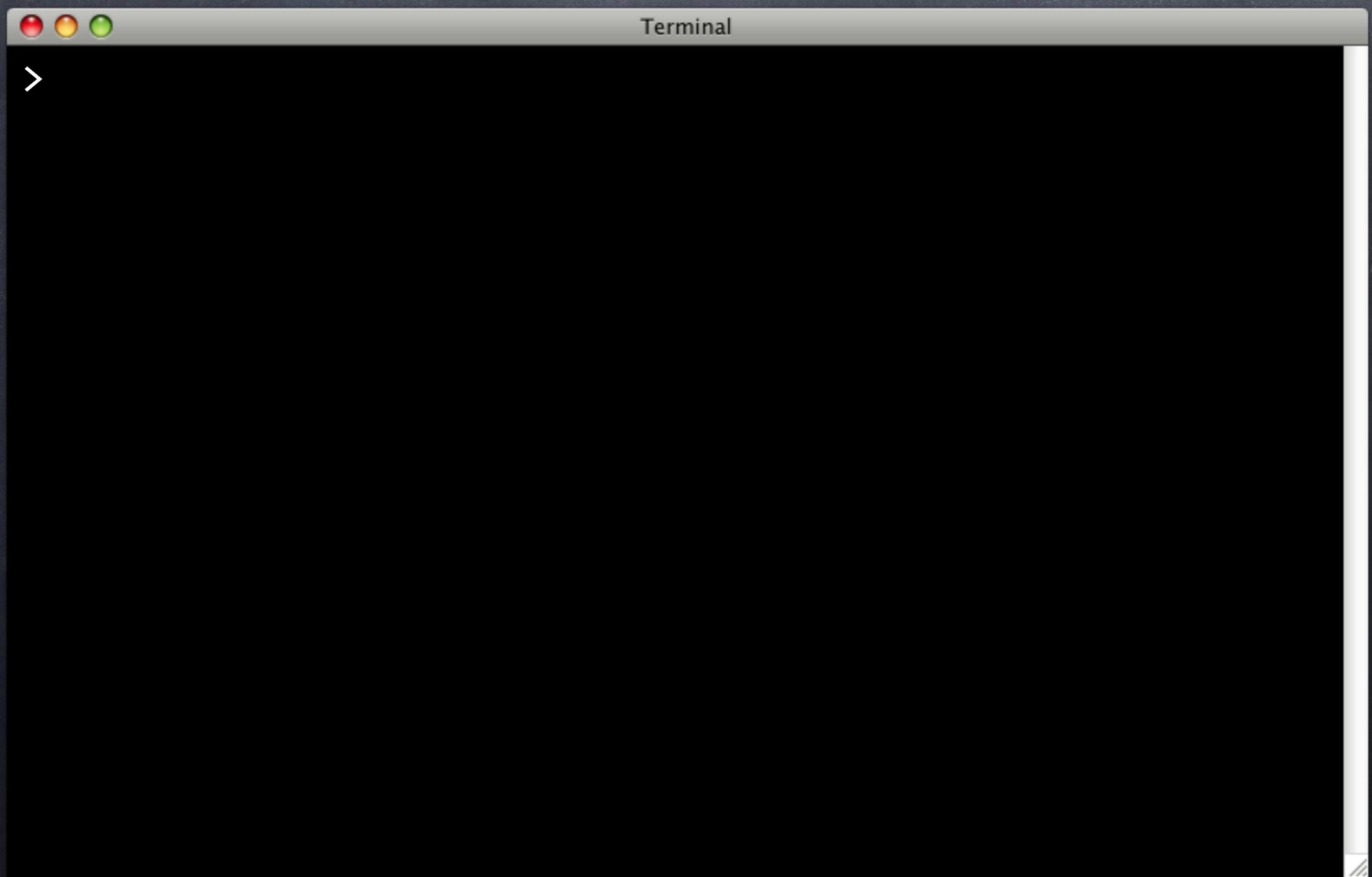
Sqitch Terminology

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

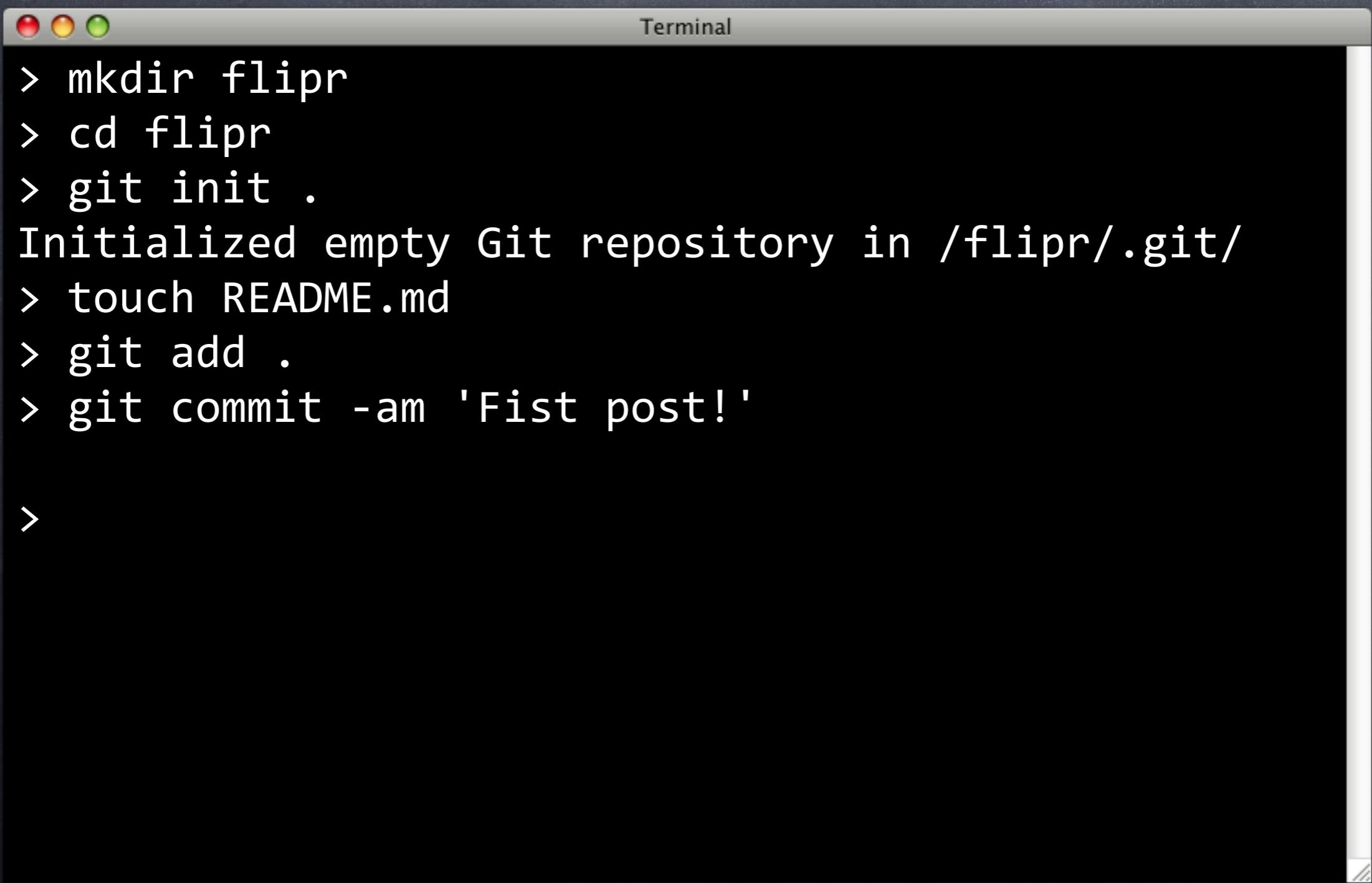
Sqitch Terminology

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

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 test/  
>
```

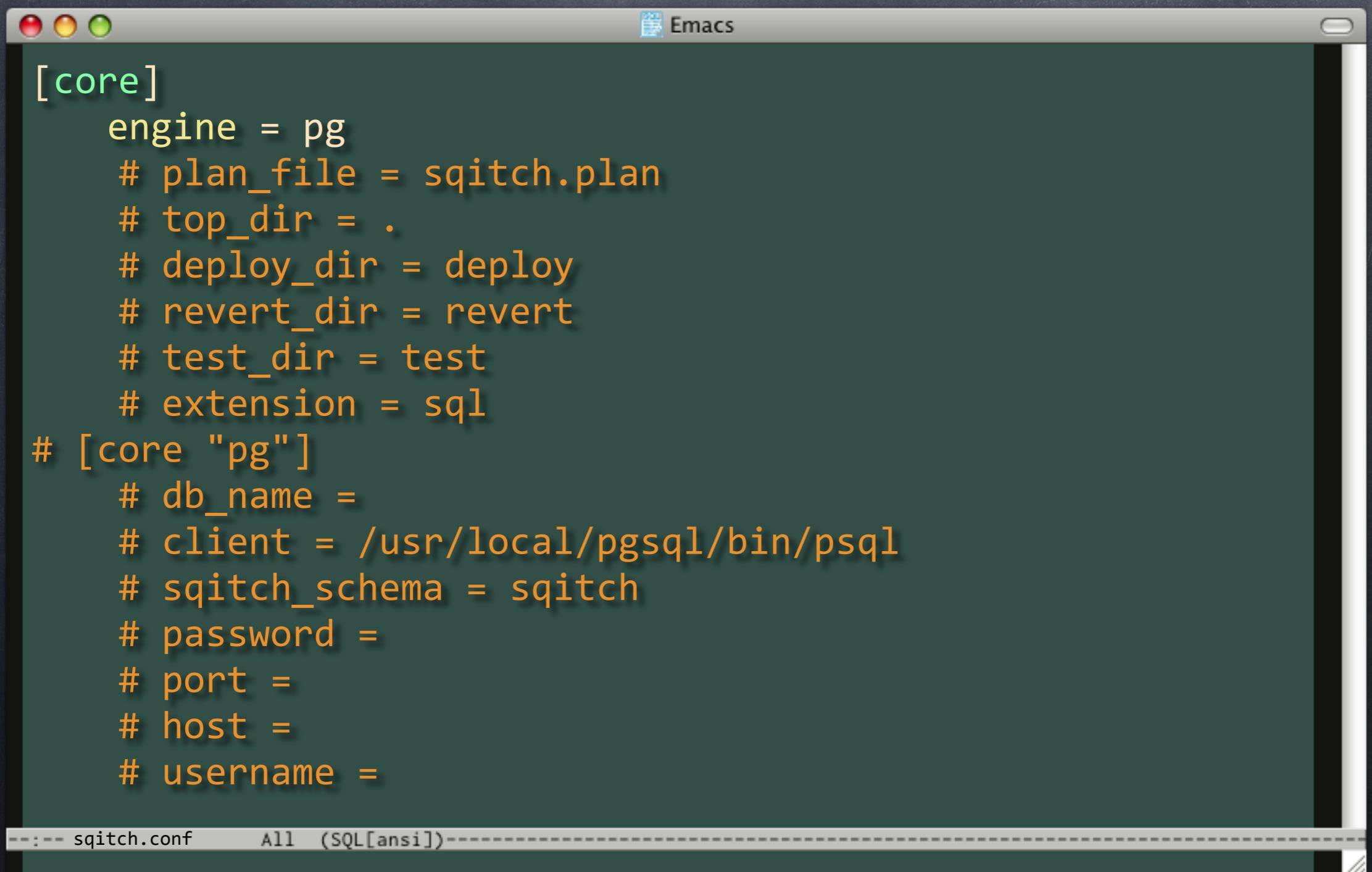
How it Works

```
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 test/  
>
```

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 test/  
> emacs sqitch.conf
```

sqitch.conf



The image shows a screenshot of an Emacs window with a dark green background and white text. The window title is "Emacs". The buffer contains a configuration file for the sqitch tool. The code is color-coded: section names are in green, and other tokens are in orange. The configuration includes settings for the core engine (pg), and detailed database connection parameters for the pg engine.

```
[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])-----

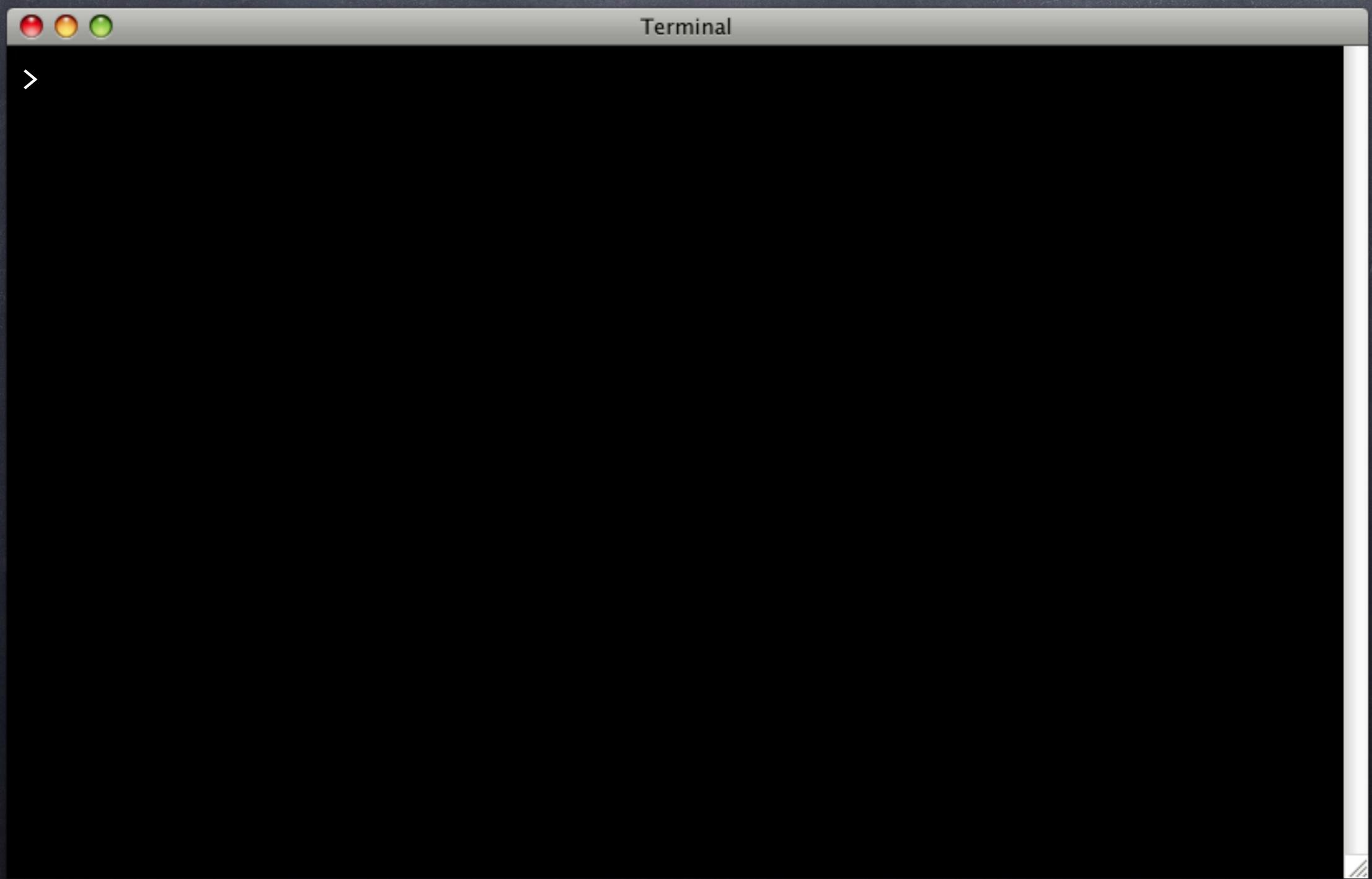
sqitch.conf



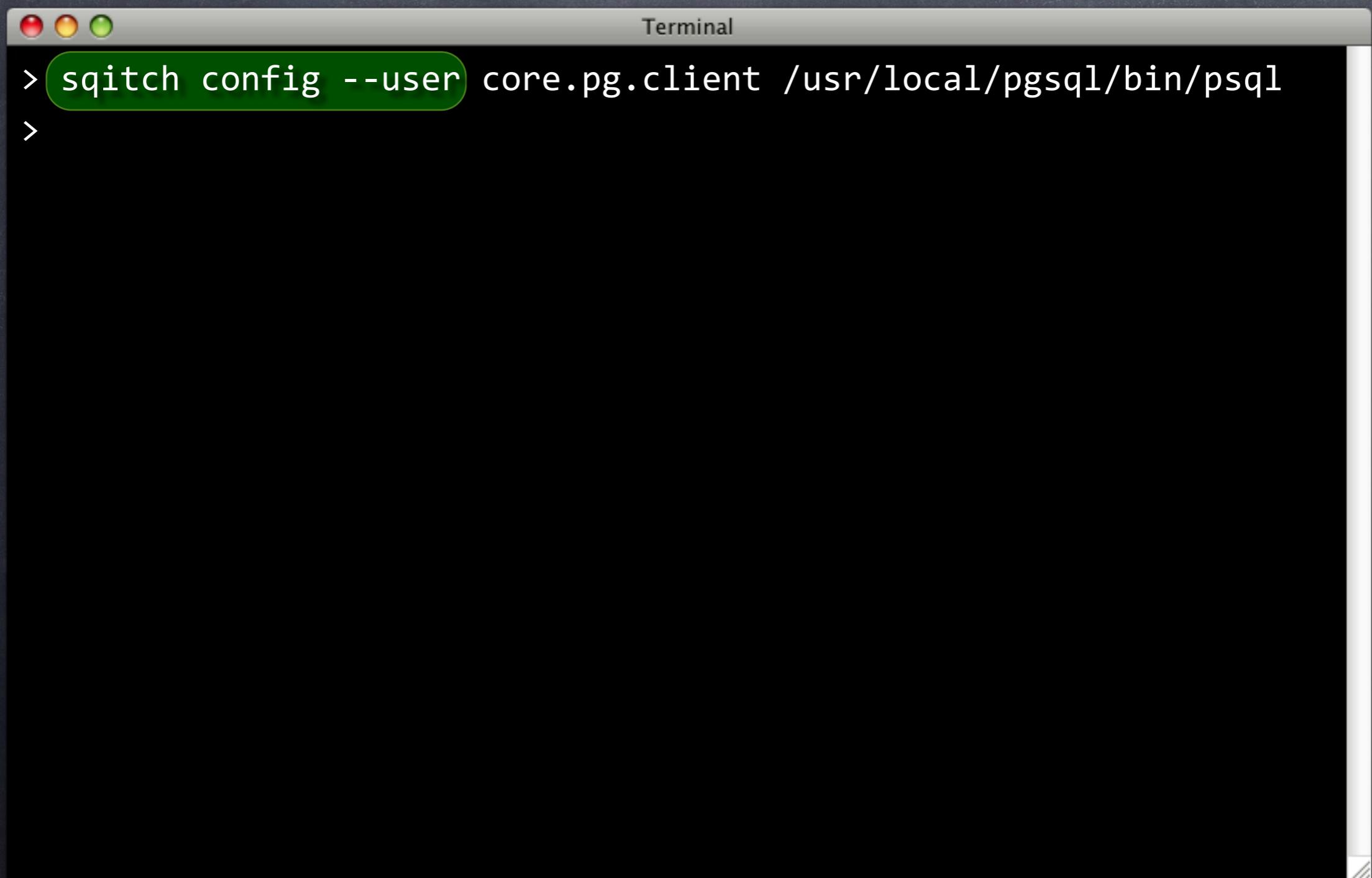
```
[core]
  engine = pg
  # plan_file =
  # top_dir =
  # deploy_dir = deploy
  # revert_dir = revert
  # test_dir = test
  # extension = sql
# [core "pg"]
  # db_name =
  # client = /usr/localpgsql/bin/psql
  # sqitch_schema = sqitch
  # password =
  # port =
  # host =
  # username =
```

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

Add User Config



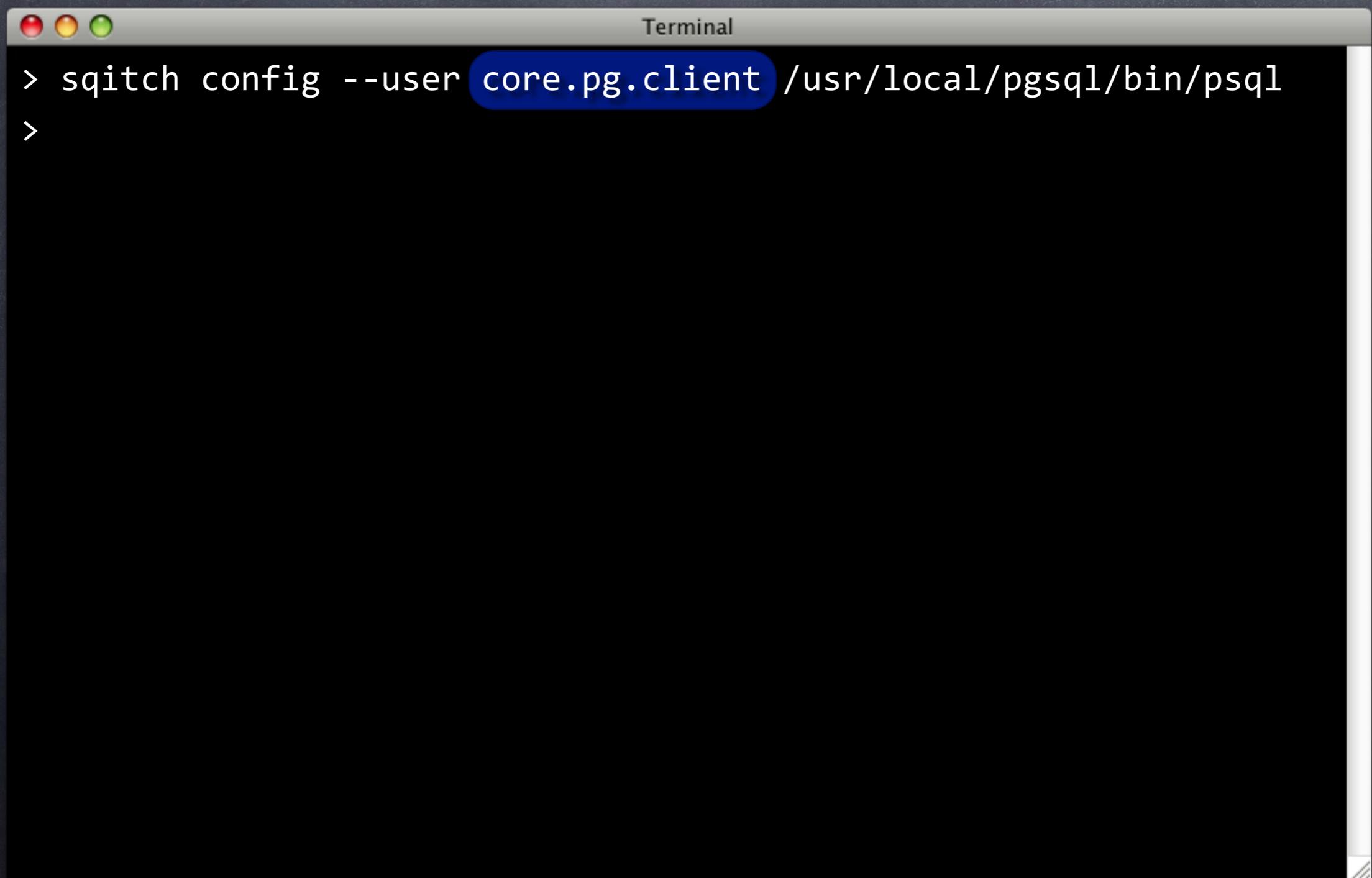
Add User Config



A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The title bar reads "Terminal". The main pane contains a command-line interface. A green rounded rectangle highlights the command "sqitch config --user core.pg.client /usr/local/pgsql/bin/psql". The command is preceded by a black '>' symbol. Below the highlighted command, there is another black '>' symbol, indicating the command has been entered but not yet run.

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

Add User Config

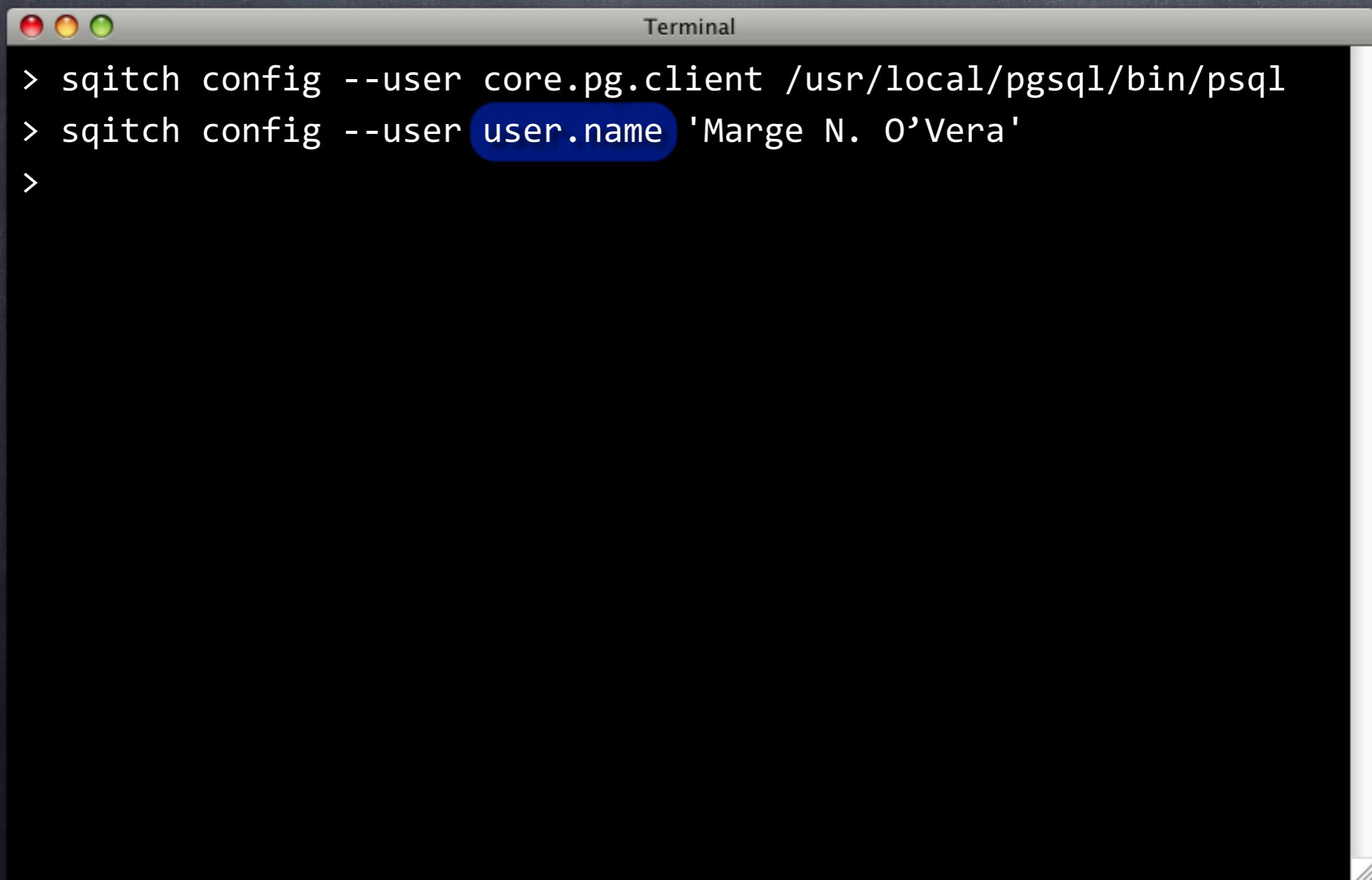


A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The title bar also displays the word "Terminal". Inside the terminal, the following command is shown:

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

The text "core.pg.client" is highlighted with a blue rounded rectangle, indicating it is the target of the user's selection.

Add User Config

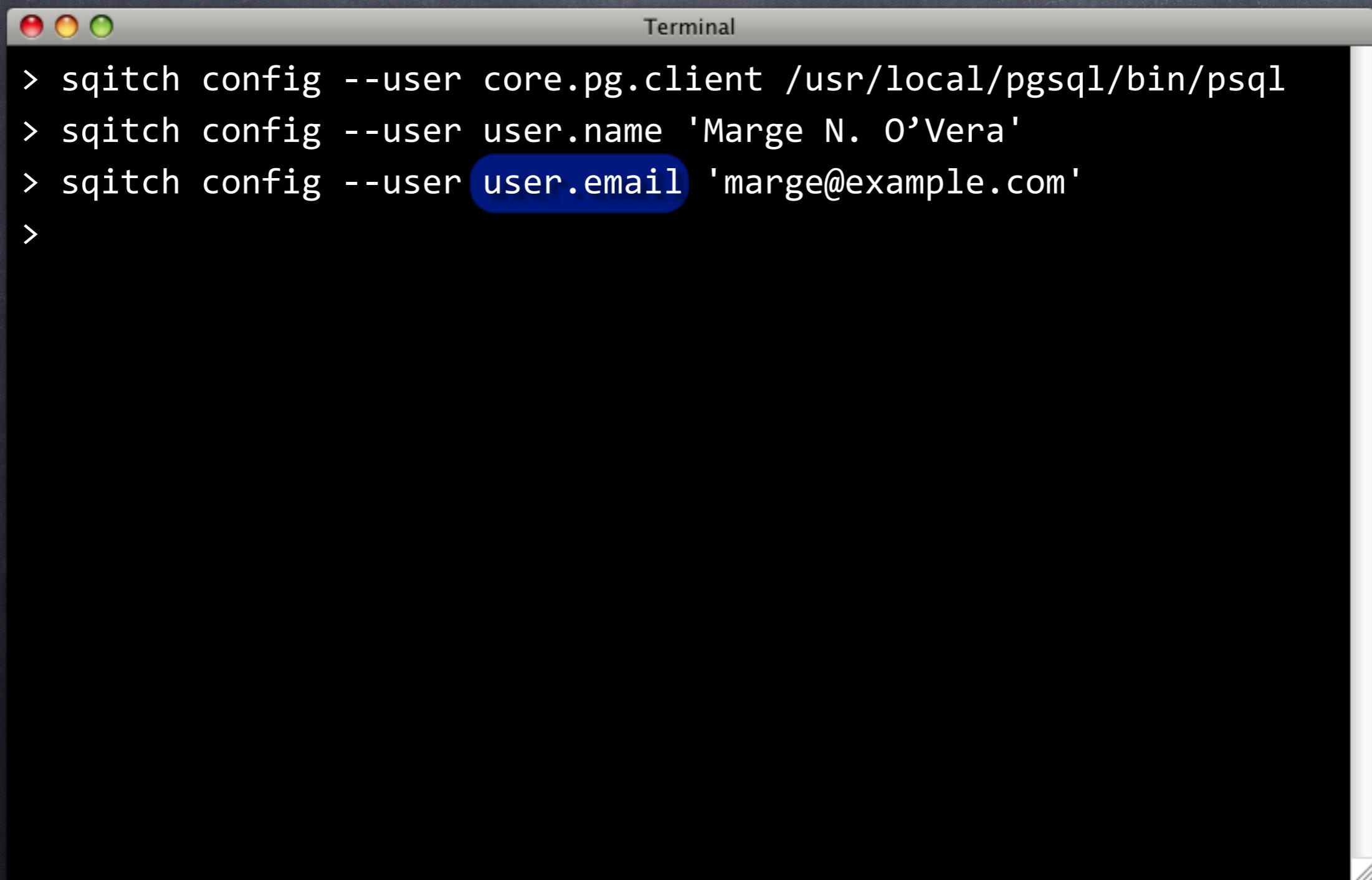


A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The terminal itself is black with white text. It contains three lines of text:

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

The word "user.name" in the second line is highlighted with a blue rounded rectangle.

Add User Config

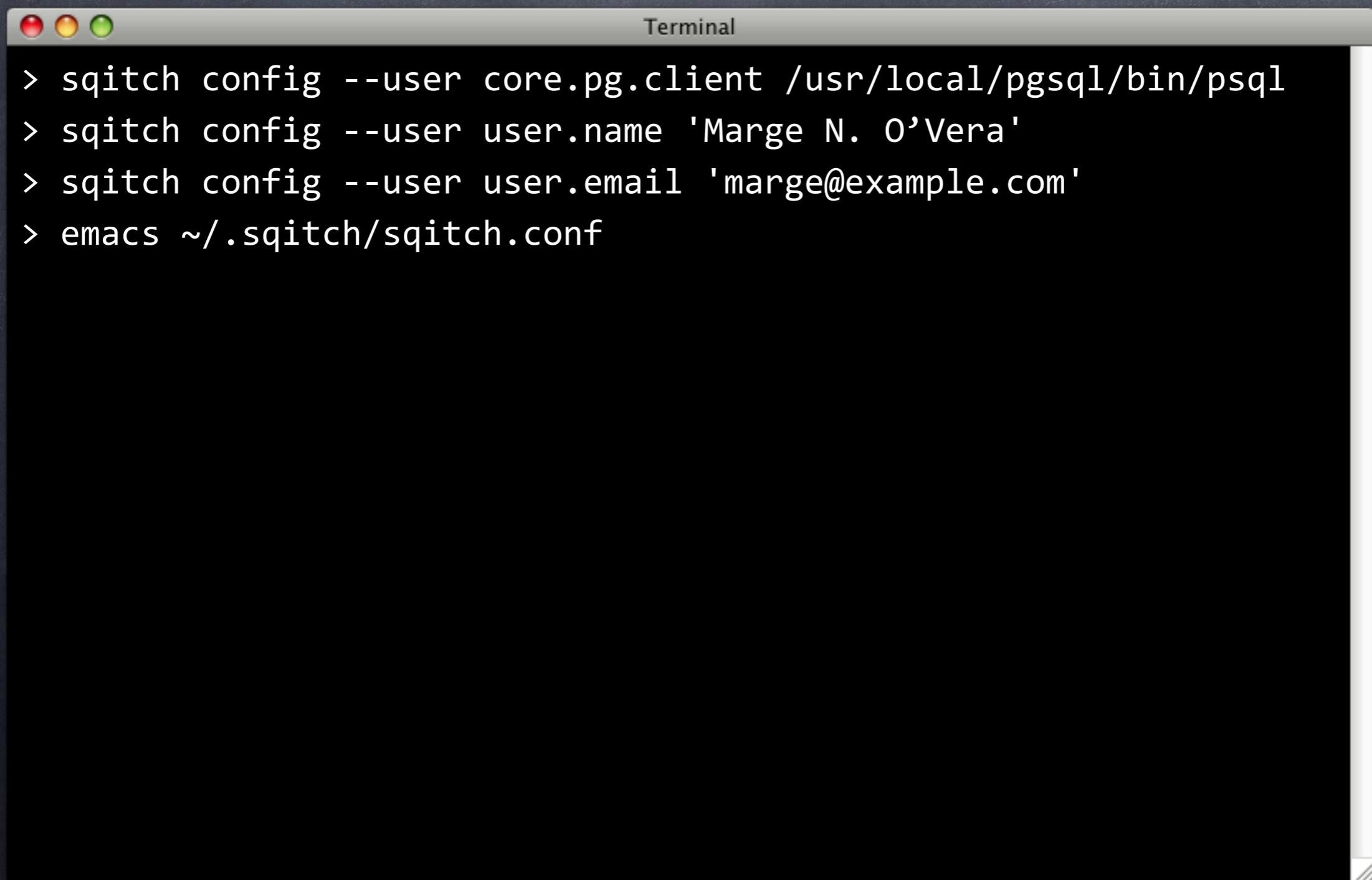


A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The title bar reads "Terminal". The main pane contains the following text:

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

The line "sqitch config --user user.email 'marge@example.com'" is highlighted with a blue rounded rectangle.

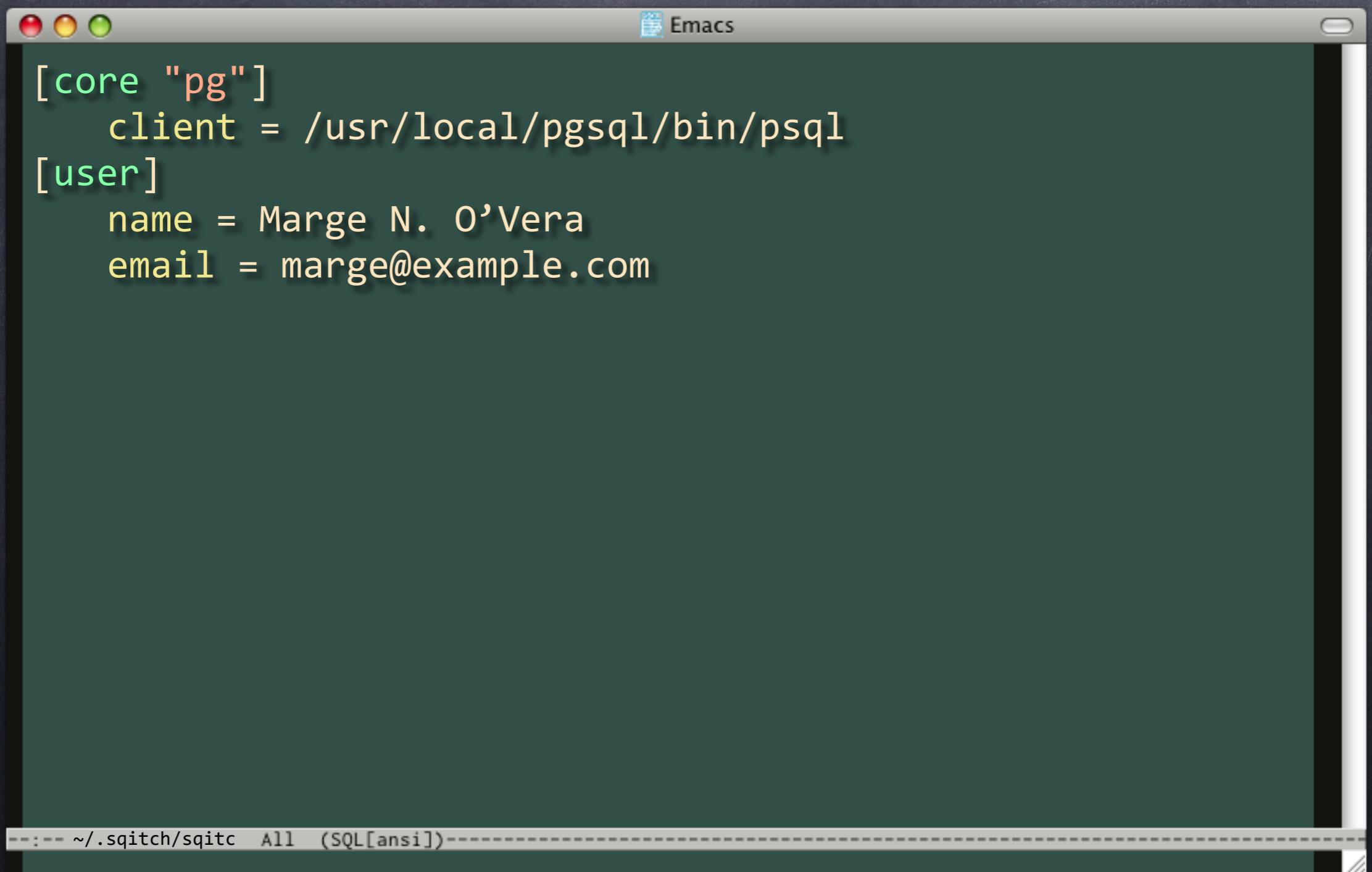
Add User Config



A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The title bar is labeled "Terminal". The main pane contains the following text:

```
> sqitch config --user core.pg.client /usr/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



The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". The buffer contains the following configuration file:

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

At the bottom of the window, there is a status bar with the text "----:--- ~/.sqitch/sqitc All (SQL[ansi])----".

~/.sqitch/sqitch.conf



The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". The buffer contains the following configuration file:

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

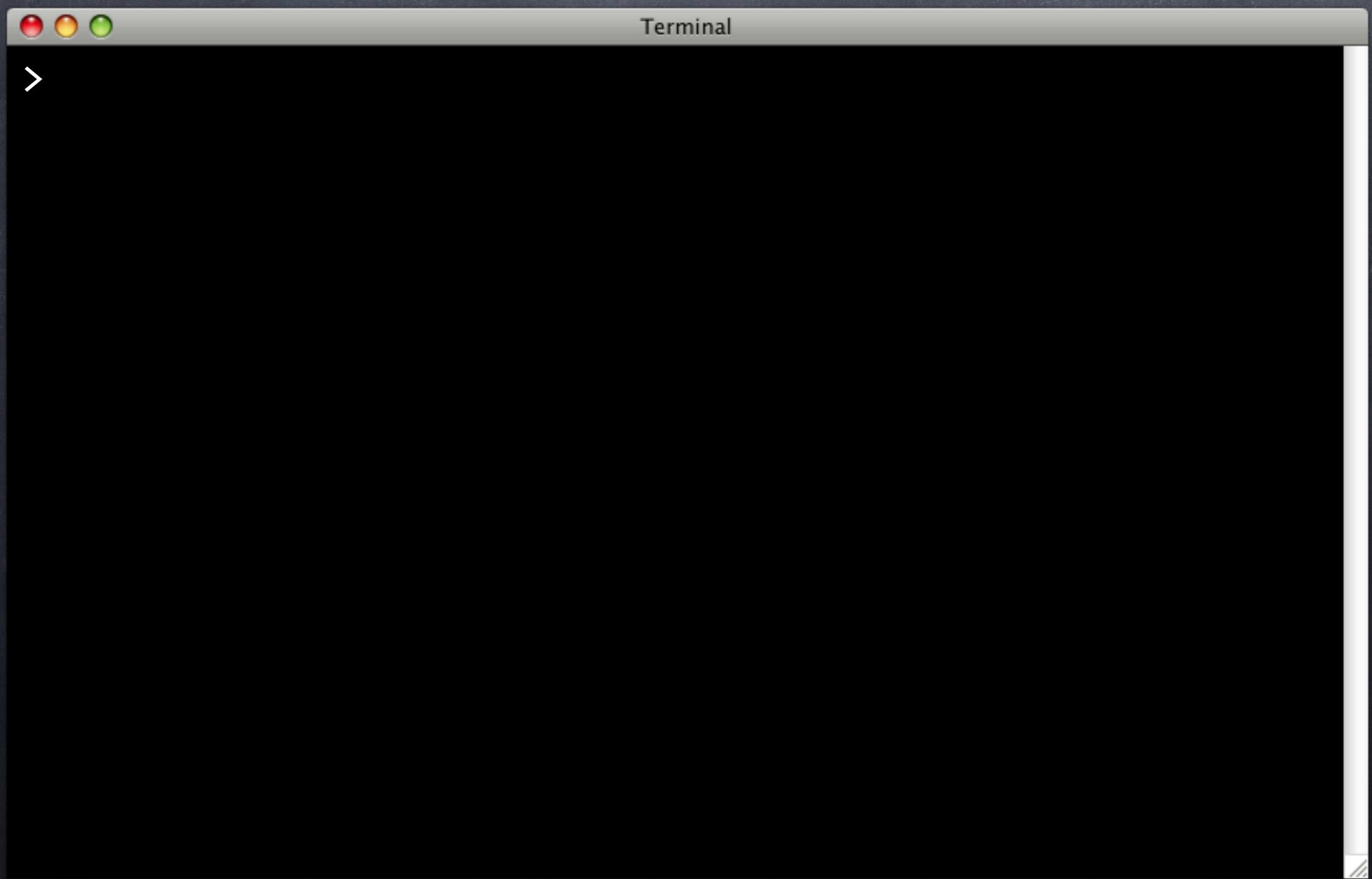
Below the window, a command-line interface shows:

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

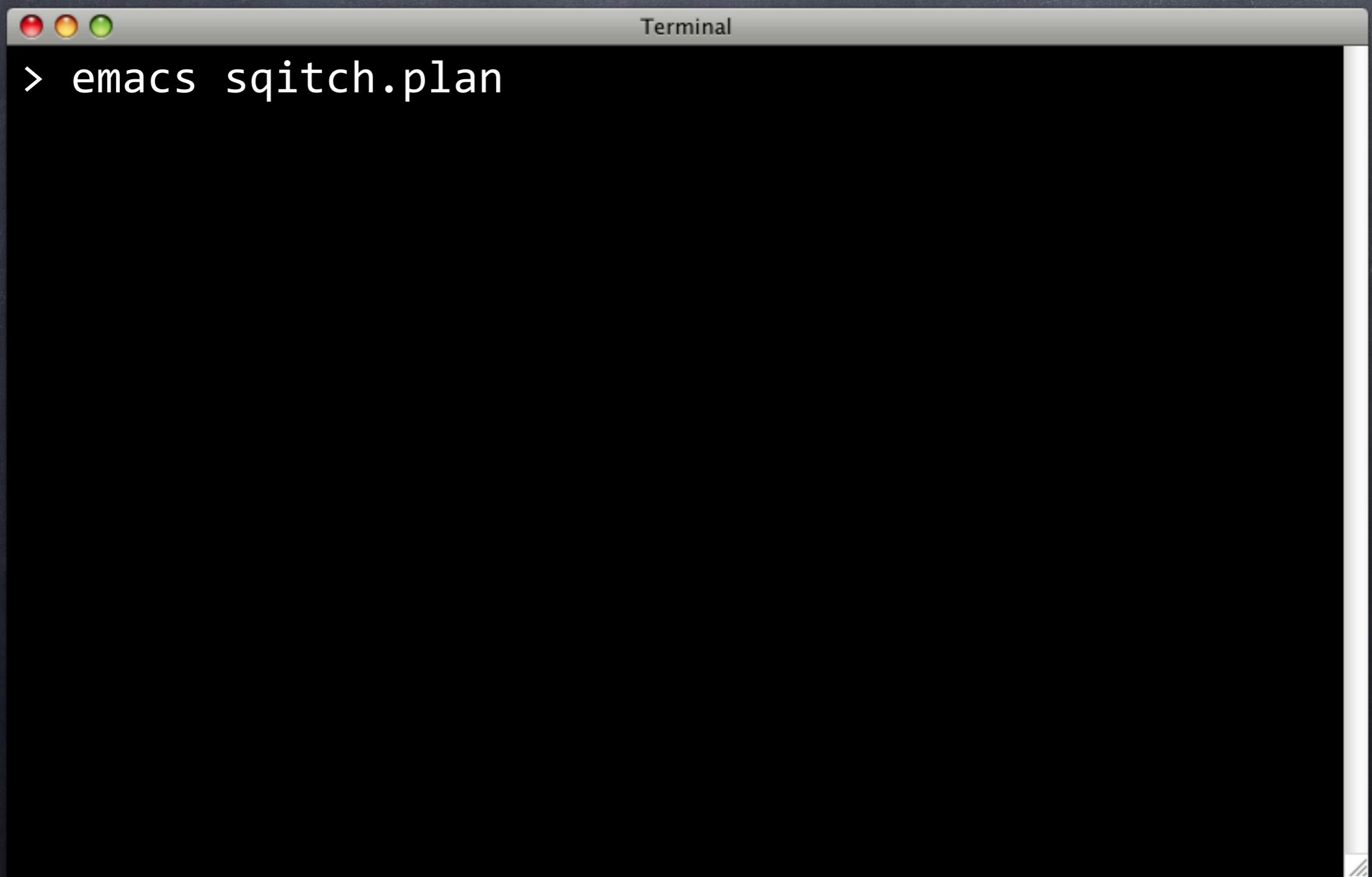
Large white text overlaid on the bottom right of the window reads:

Good for
all projects

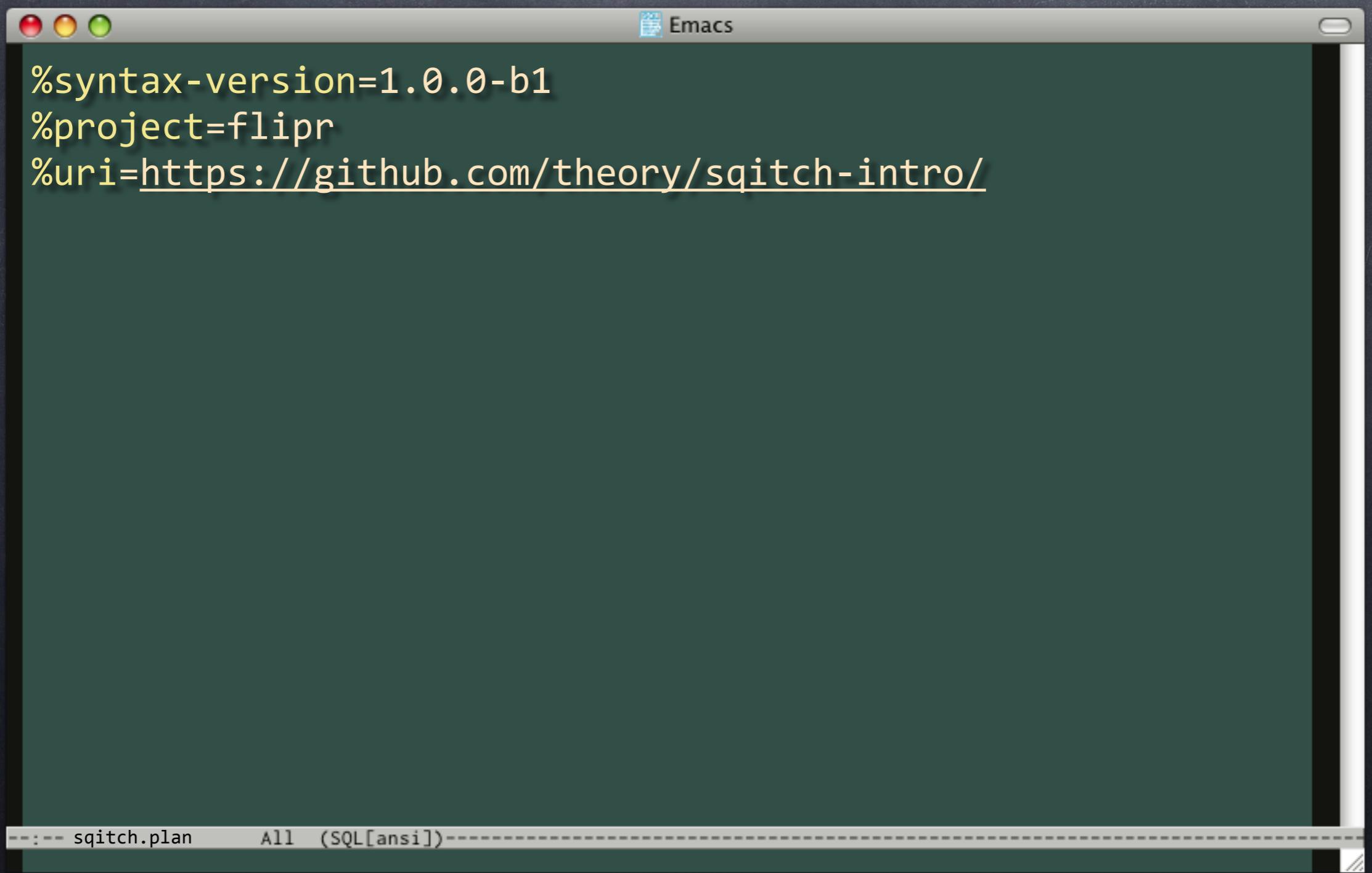
What's the Plan Man?



What's the Plan Man?



sqitch.plan



The screenshot shows a window titled "Emacs" with a dark green background. The window contains the following text:

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

The text is in white, with the "uri" line containing a URL that is underlined. The window has a standard OS X style title bar with red, yellow, and green buttons. At the bottom, there is a status bar with the text "sqitch.plan All (SQL[ansi])".

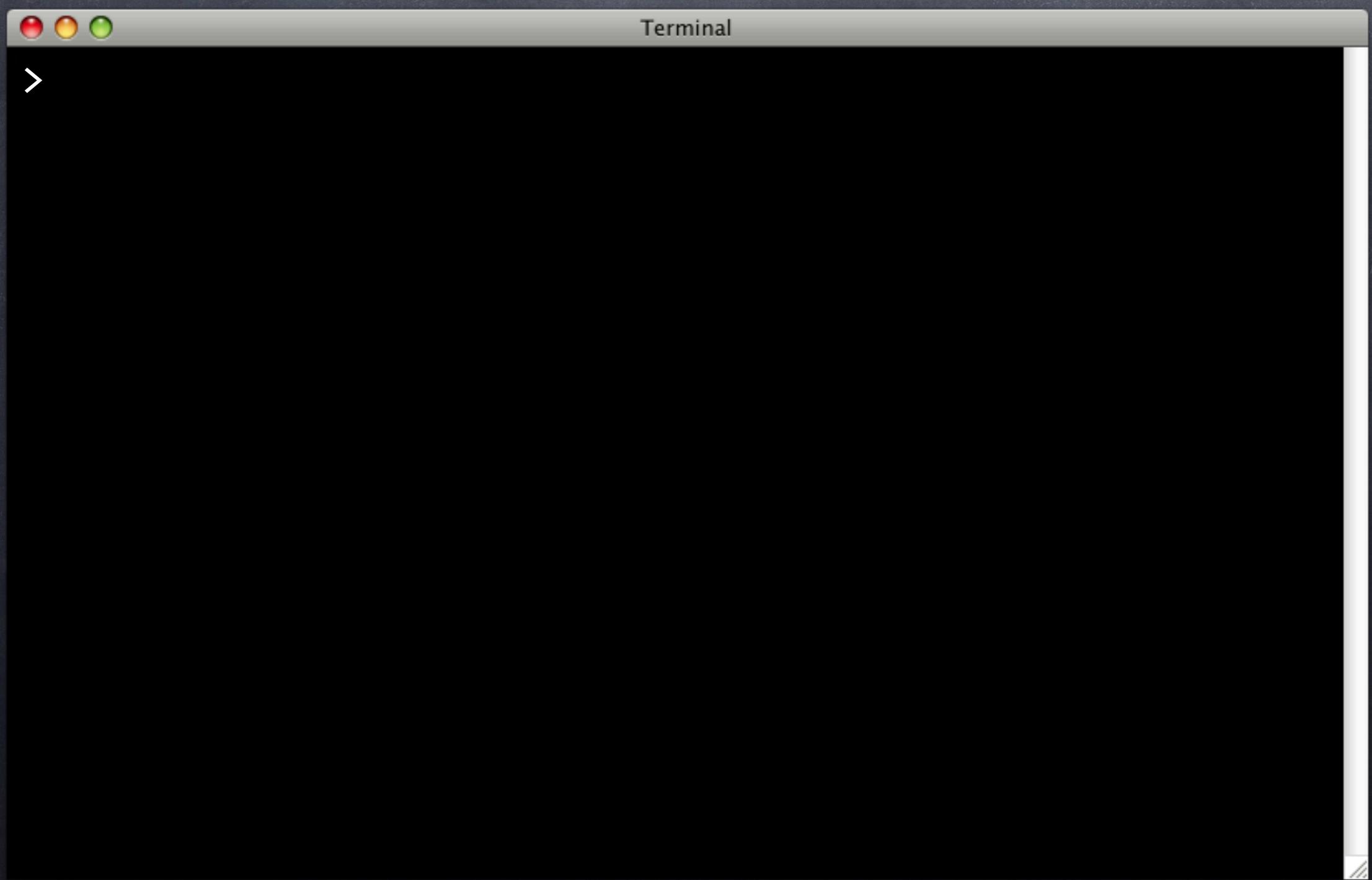
sqitch.plan

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

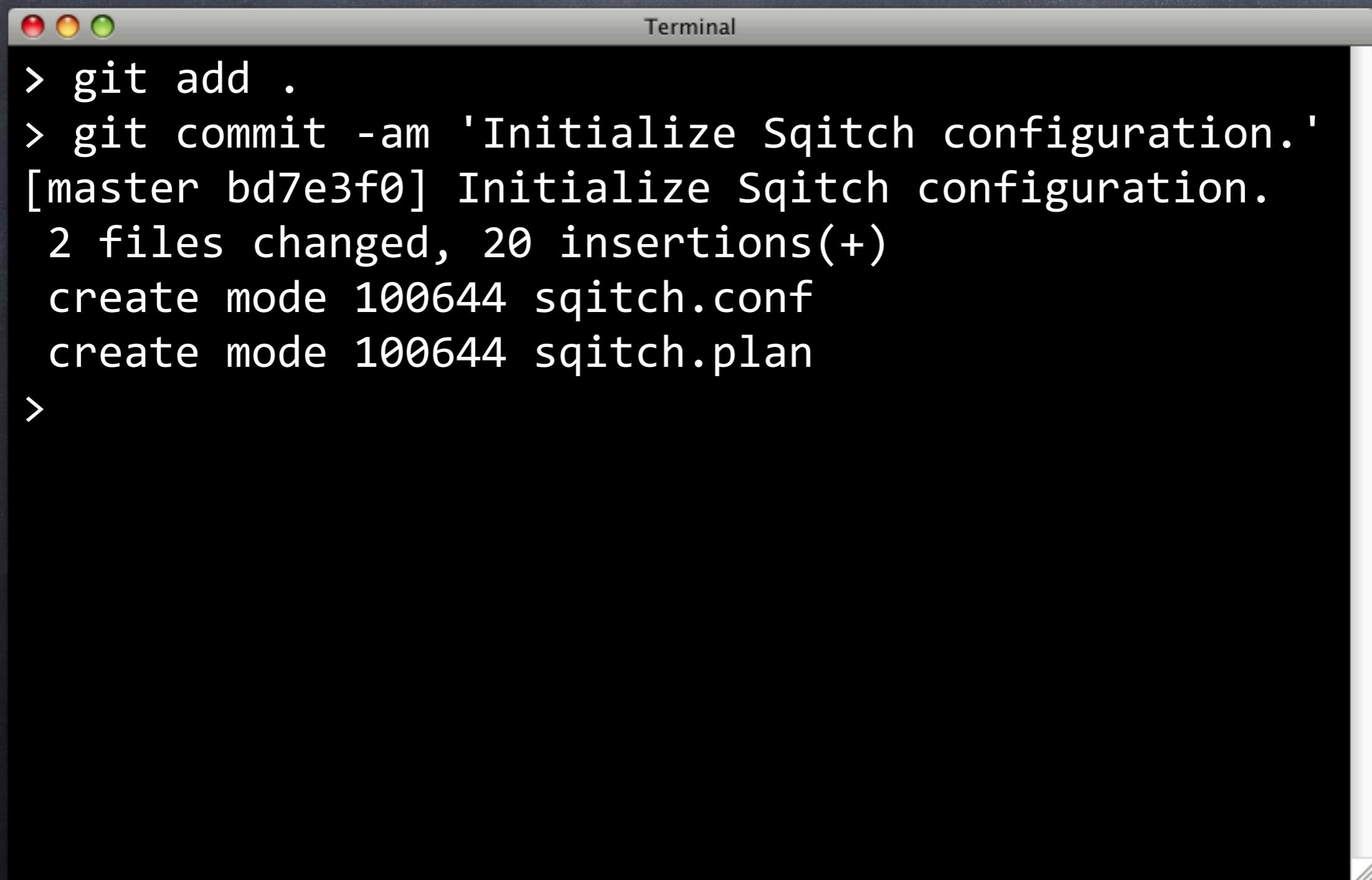
Identified

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

Make It So

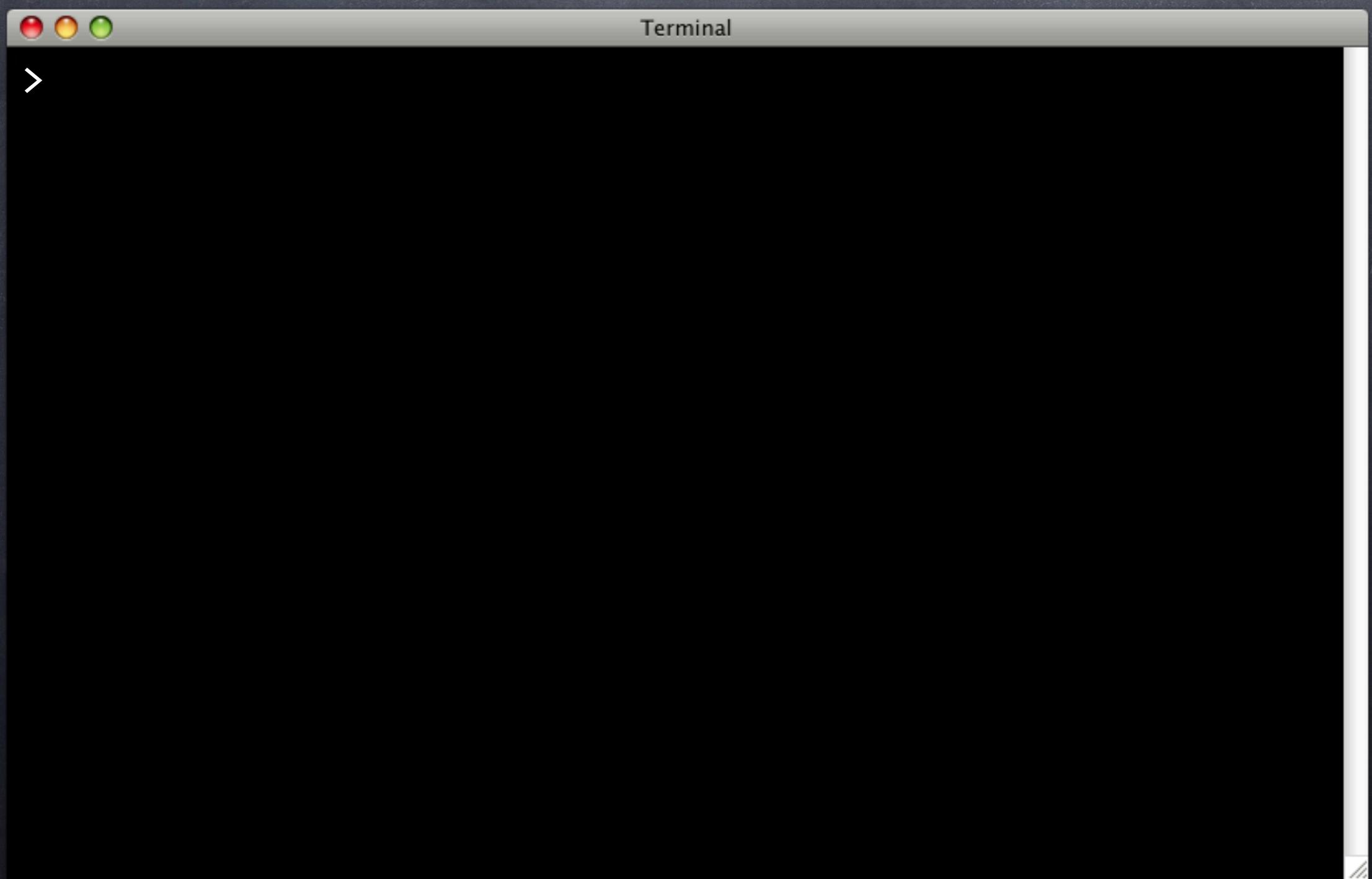


Make It So

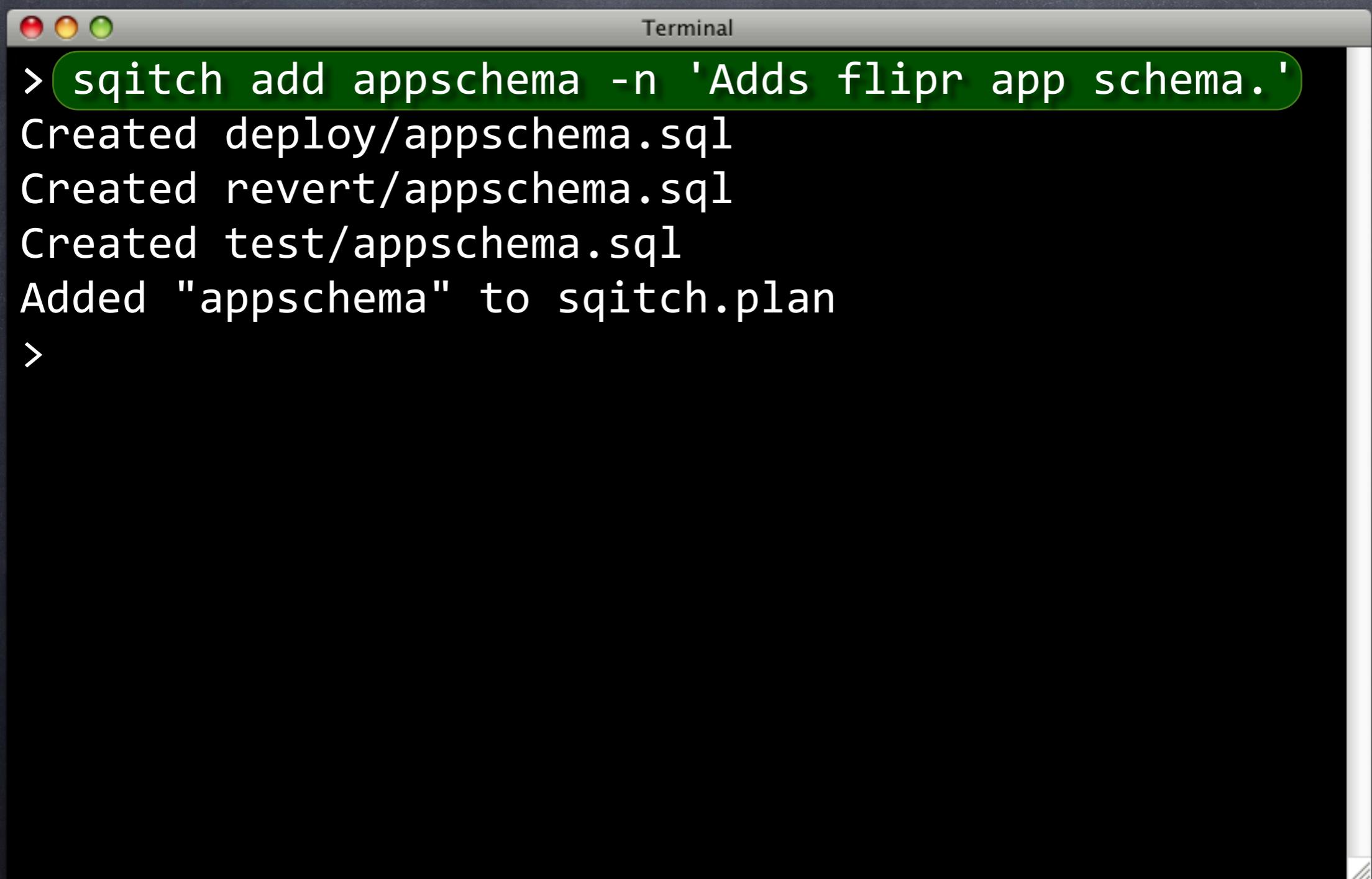


```
Terminal  
> git add .  
> git commit -am 'Initialize Sqitch configuration.'  
[master bd7e3f0] Initialize Sqitch configuration.  
 2 files changed, 20 insertions(+)  
 create mode 100644 sqitch.conf  
 create mode 100644 sqitch.plan  
>
```

First Deployment



First Deployment



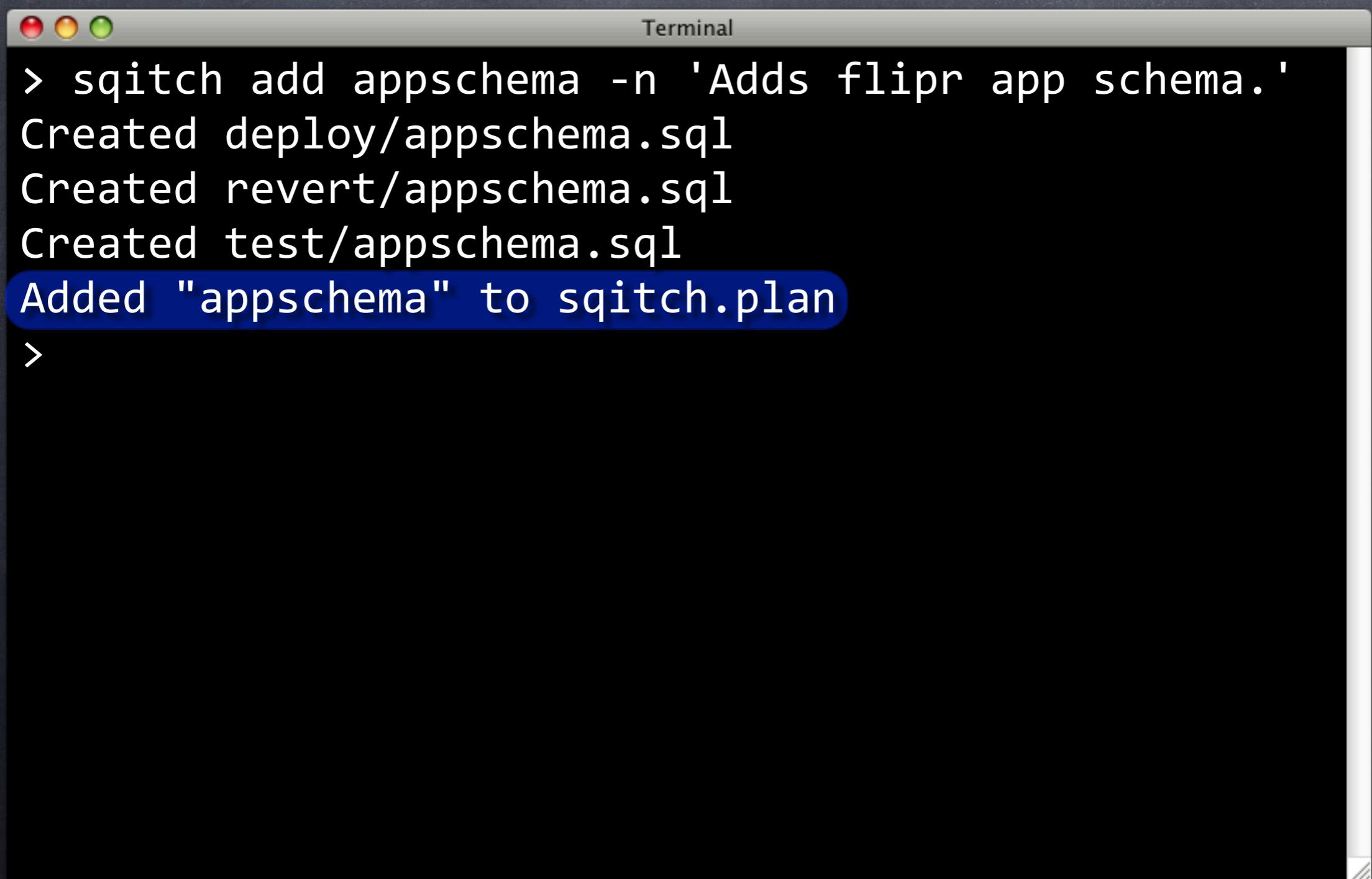
A screenshot of a Mac OS X Terminal window titled "Terminal". The window has a standard title bar with red, yellow, and green close buttons. The main pane shows the following text:

```
> 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  
>
```

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  
>
```

First Deployment

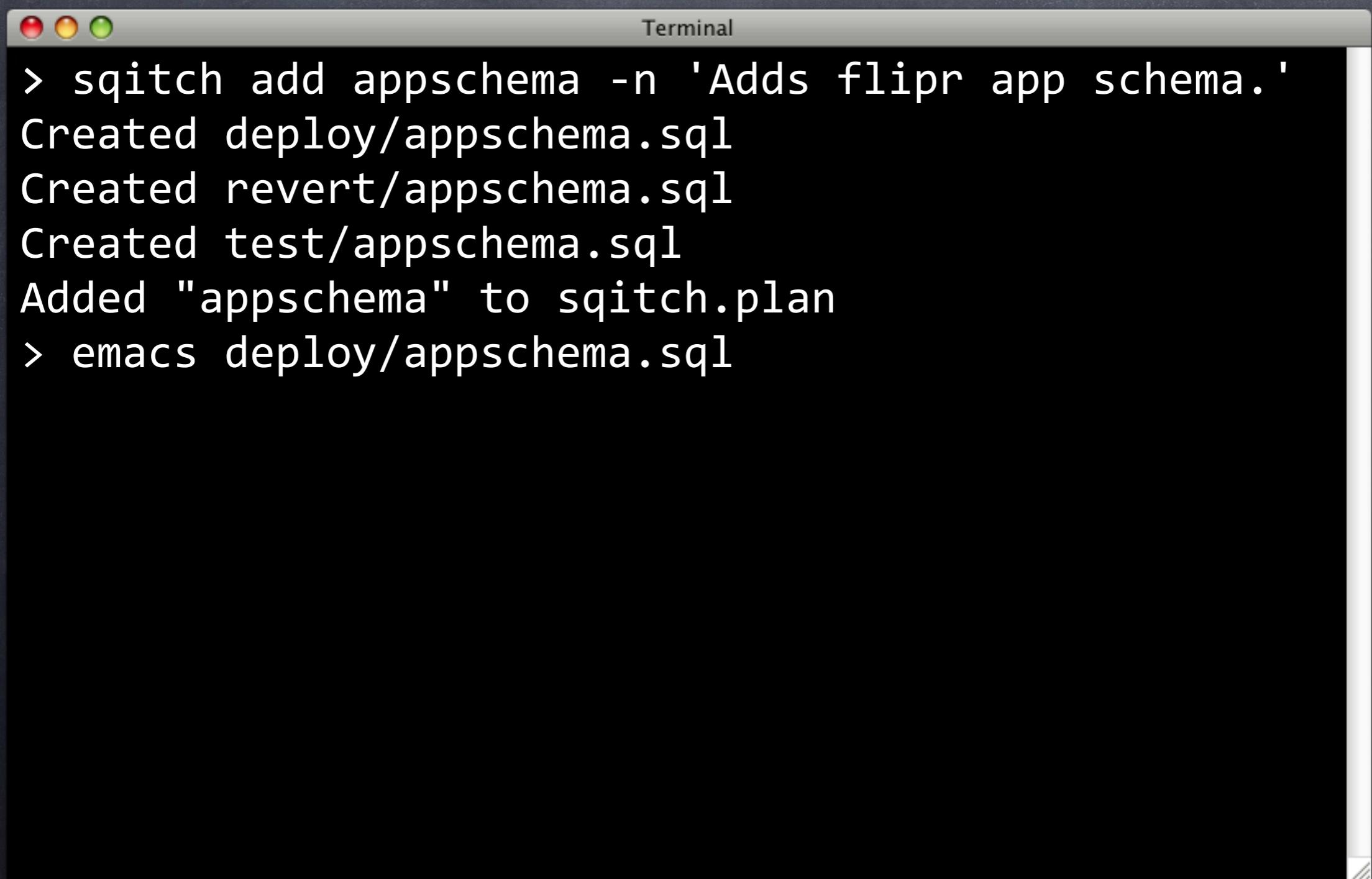


A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the following text:

```
> 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  
>
```

The line "Added "appschema" to sqitch.plan" is highlighted with a blue rounded rectangle.

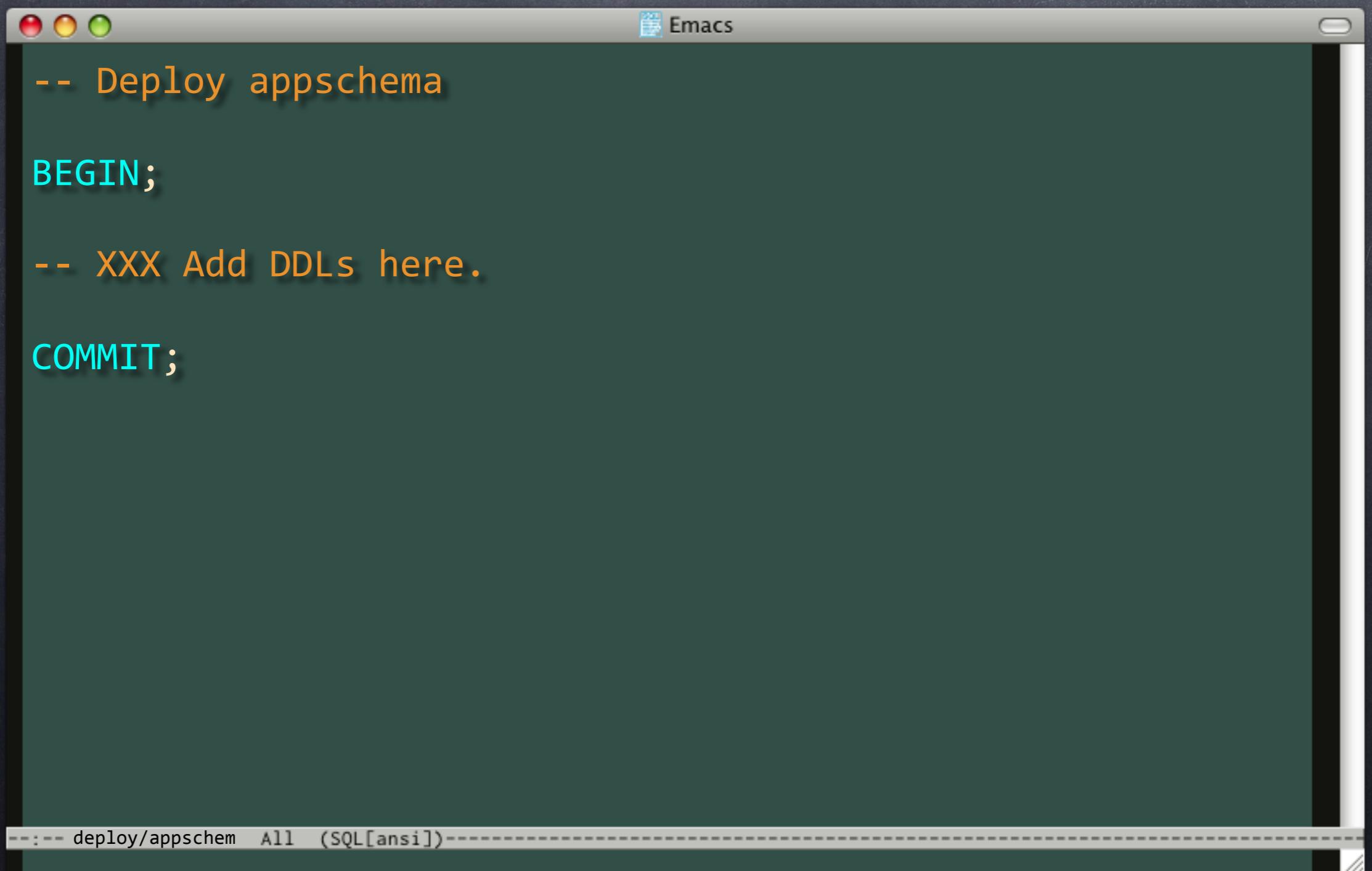
First Deployment



A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The title bar reads "Terminal". The main pane contains the following text:

```
> 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
```

deploy/appschema.sql

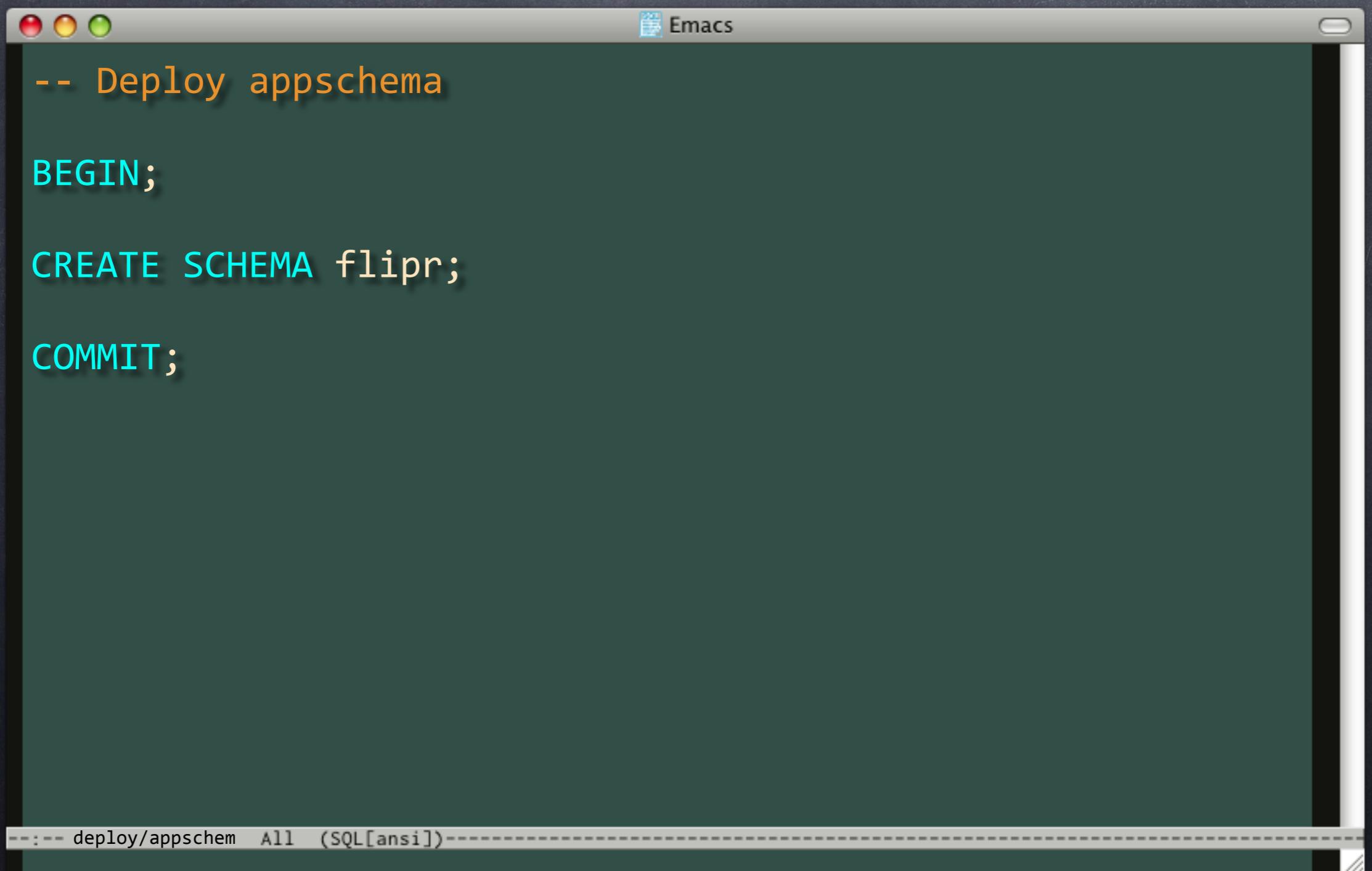


The image shows a screenshot of an Emacs window with a dark green background. The window title is "Emacs". The code inside the window is as follows:

```
-- Deploy appschema  
  
BEGIN;  
  
-- XXX Add DDLs here.  
  
COMMIT;
```

At the bottom of the window, there is a status bar with the text "---- deploy/appschema All (SQL[ansi])----".

deploy/appschema.sql



The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". The buffer contains the following SQL code:

```
-- Deploy appschema

BEGIN;

CREATE SCHEMA flipr;

COMMIT;
```

At the bottom of the buffer, there is a status line with the text "---- deploy/appschema All (SQL[ansi])-----".

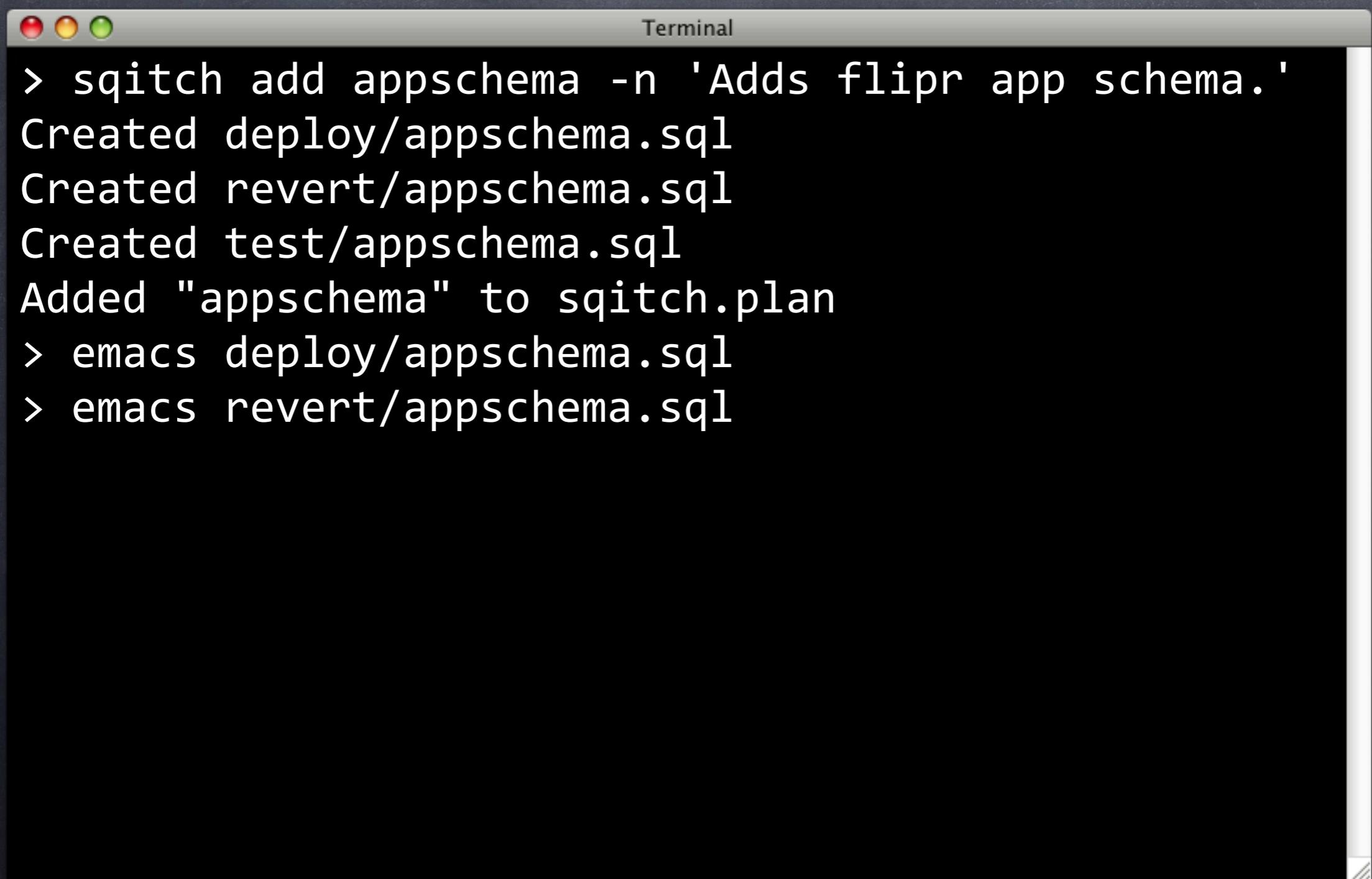
First Deployment



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  
>
```

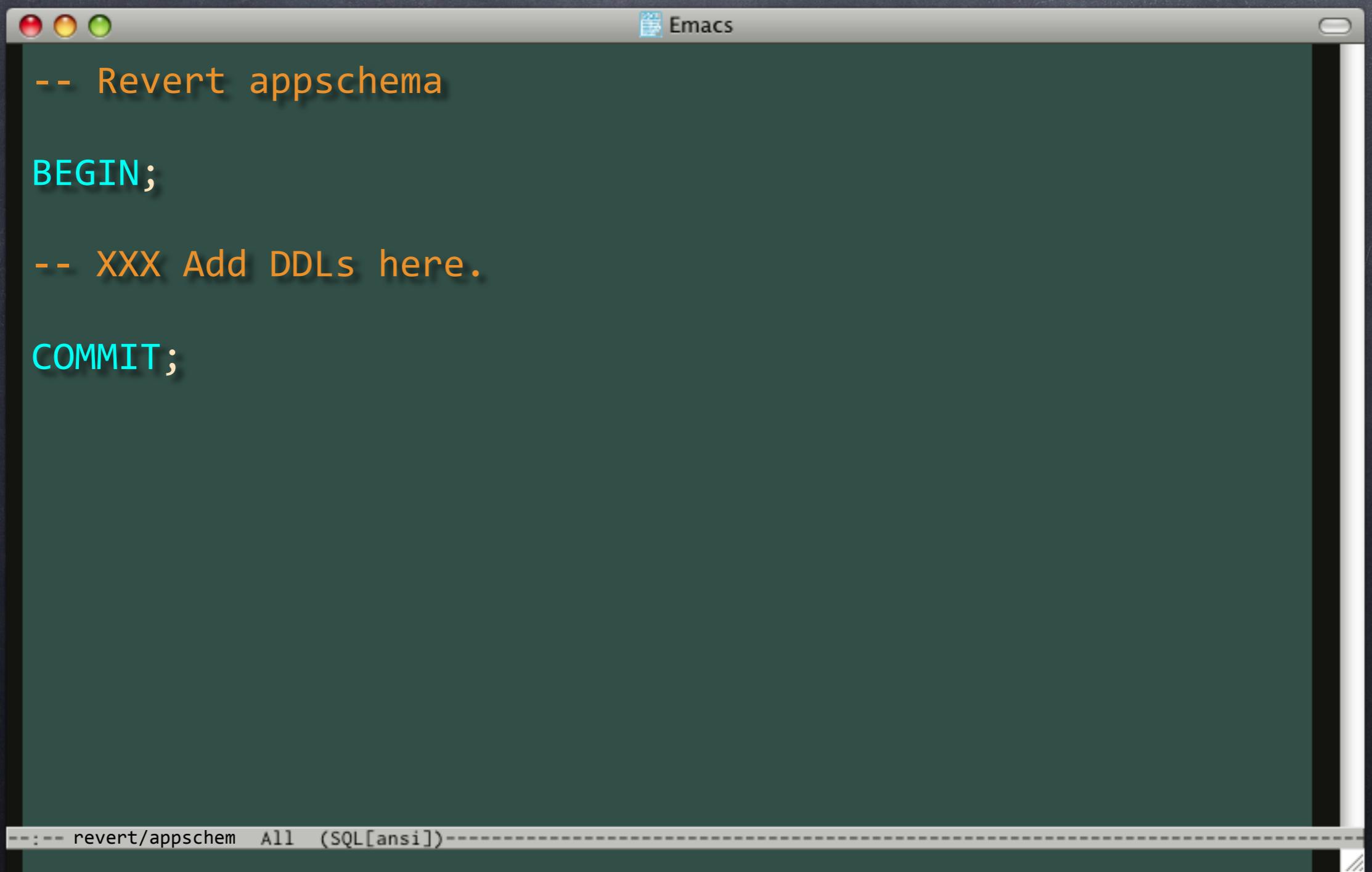
First Deployment



A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The title bar reads "Terminal". The main pane contains the following text:

```
> 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/apps schema.sql



The image shows a screenshot of an Emacs window with a dark green background. The window title is "Emacs". The buffer contains the following SQL code:

```
-- Revert appschema

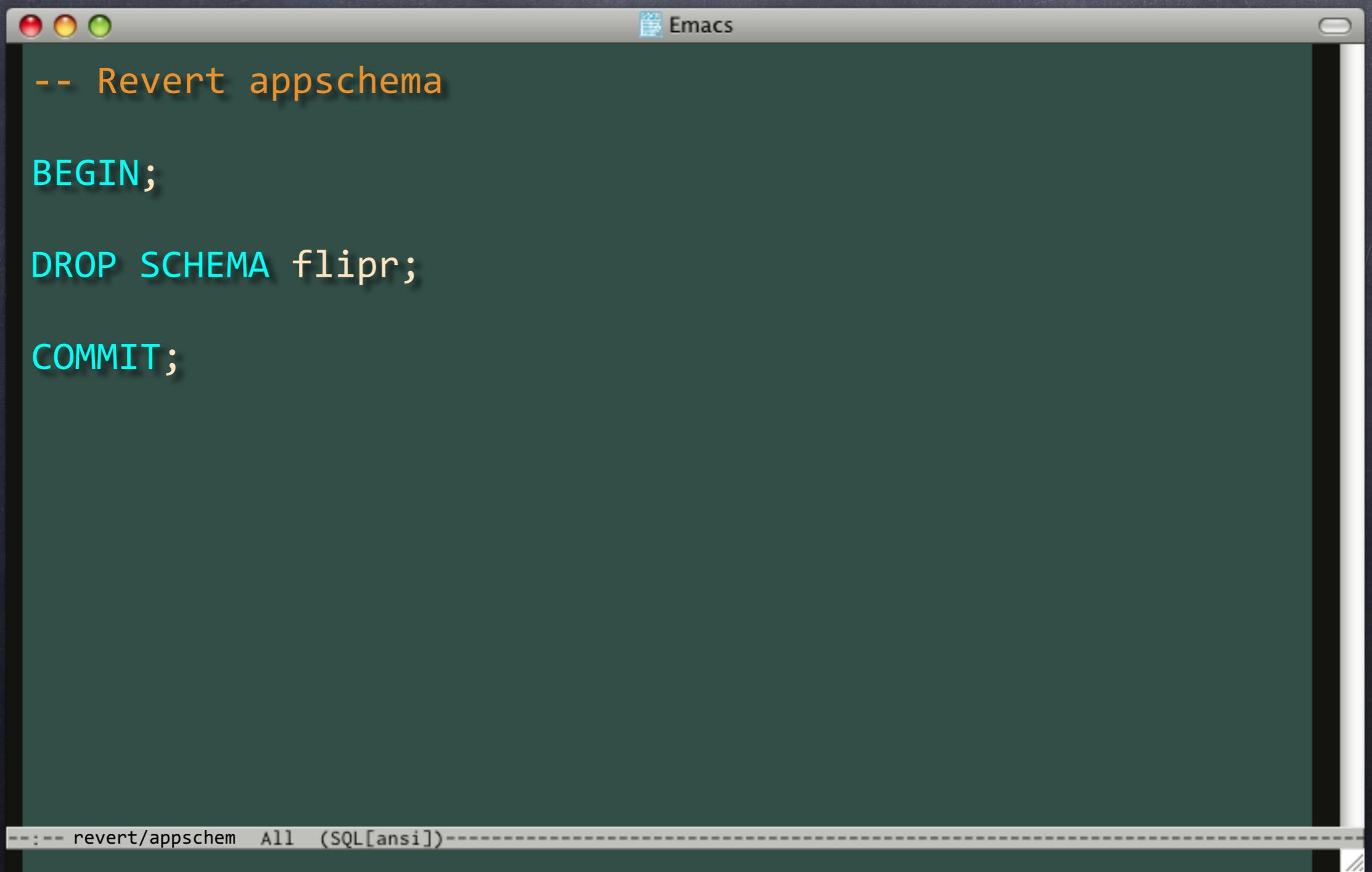
BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

At the bottom of the window, there is a status bar with the text "---- revert/appschem All (SQL[ansi])-----".

revert/apps schema.sql



The image shows a screenshot of an Emacs window with a dark green background. The window title is "Emacs". The buffer contains the following SQL code:

```
-- Revert appschema

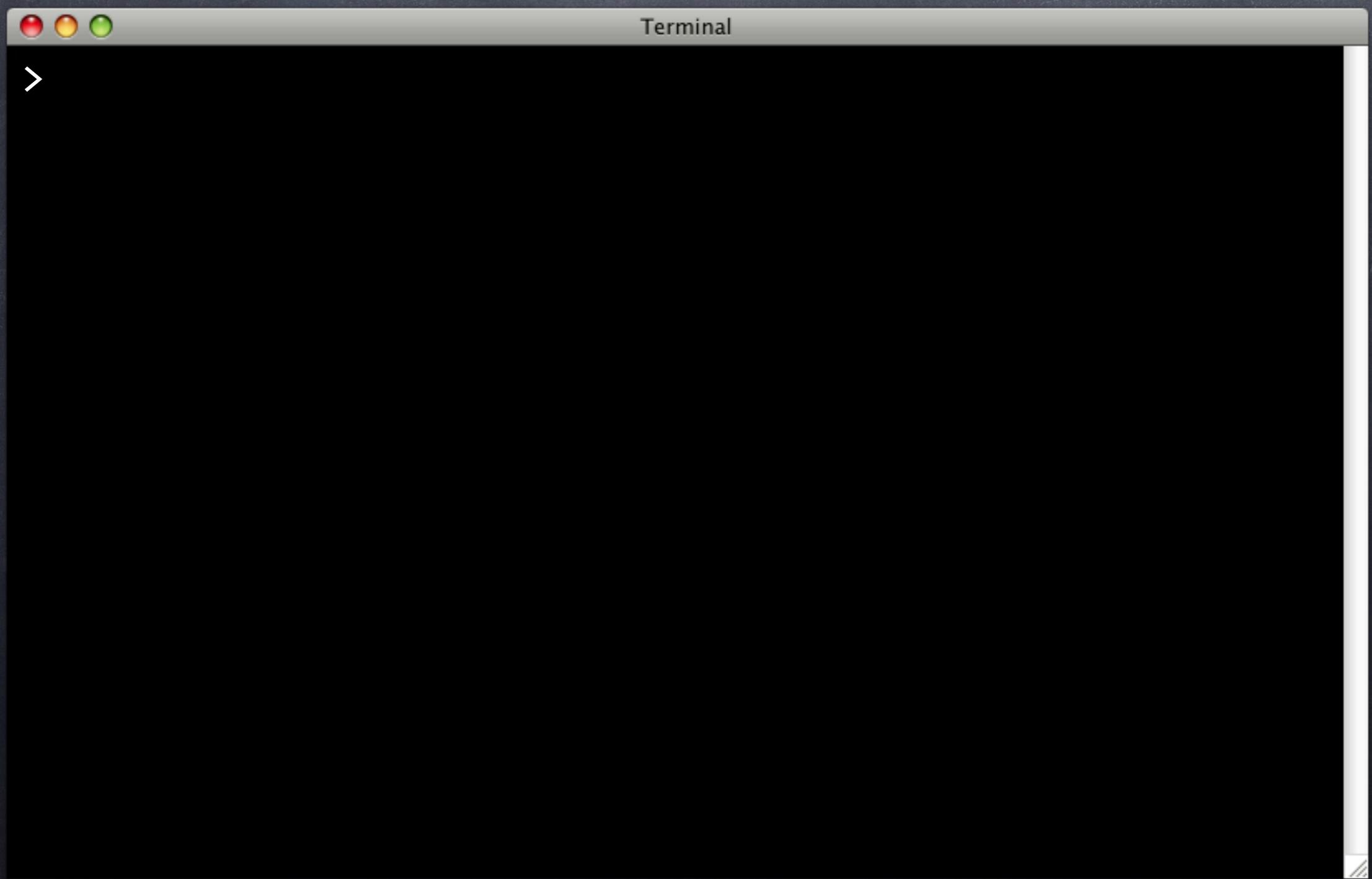
BEGIN;

DROP SCHEMA flipr;

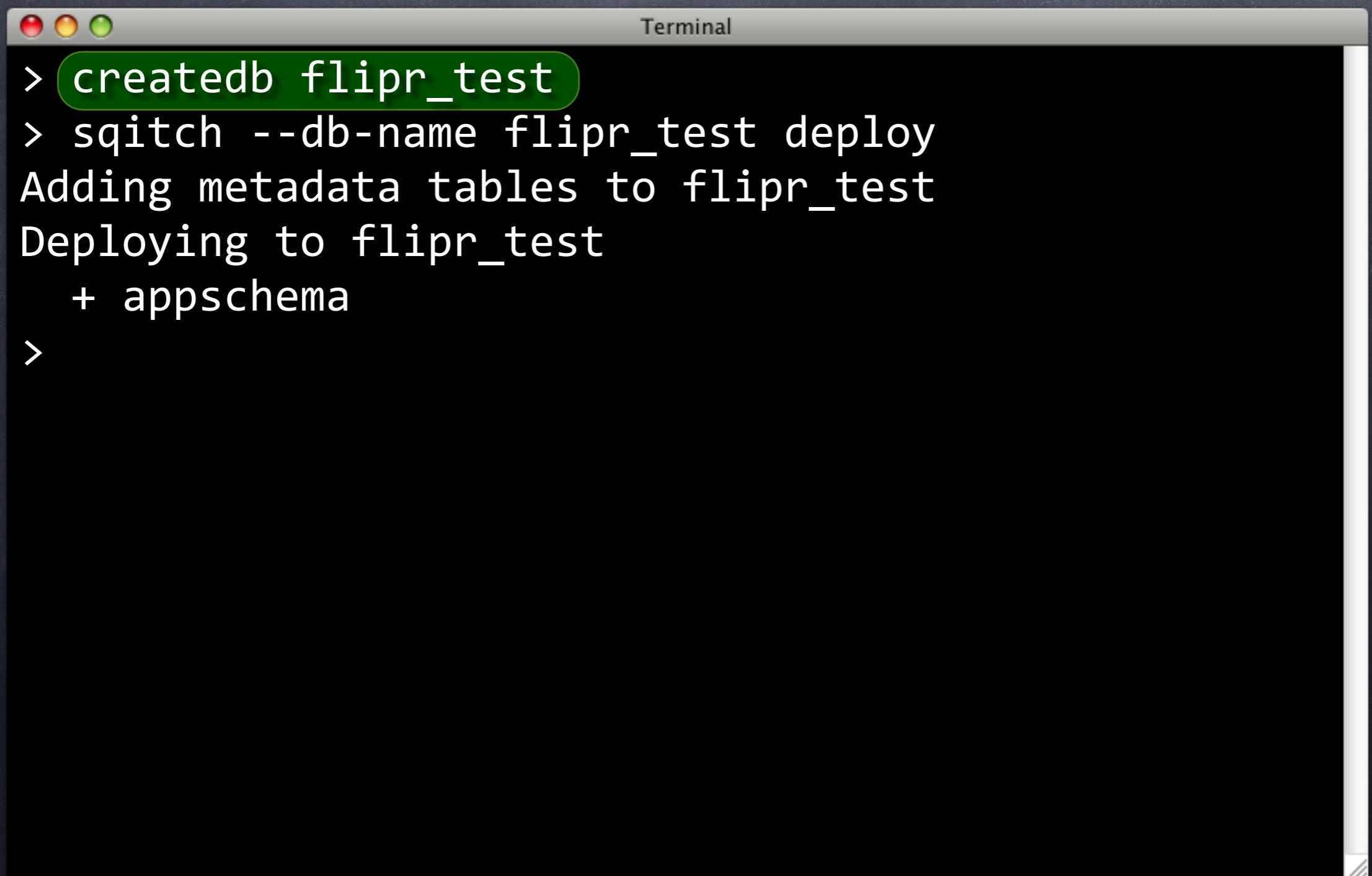
COMMIT;
```

At the bottom of the window, there is a status bar with the text "---- revert/appschem All (SQL[ansi])-----".

Make it So!



Make it So!

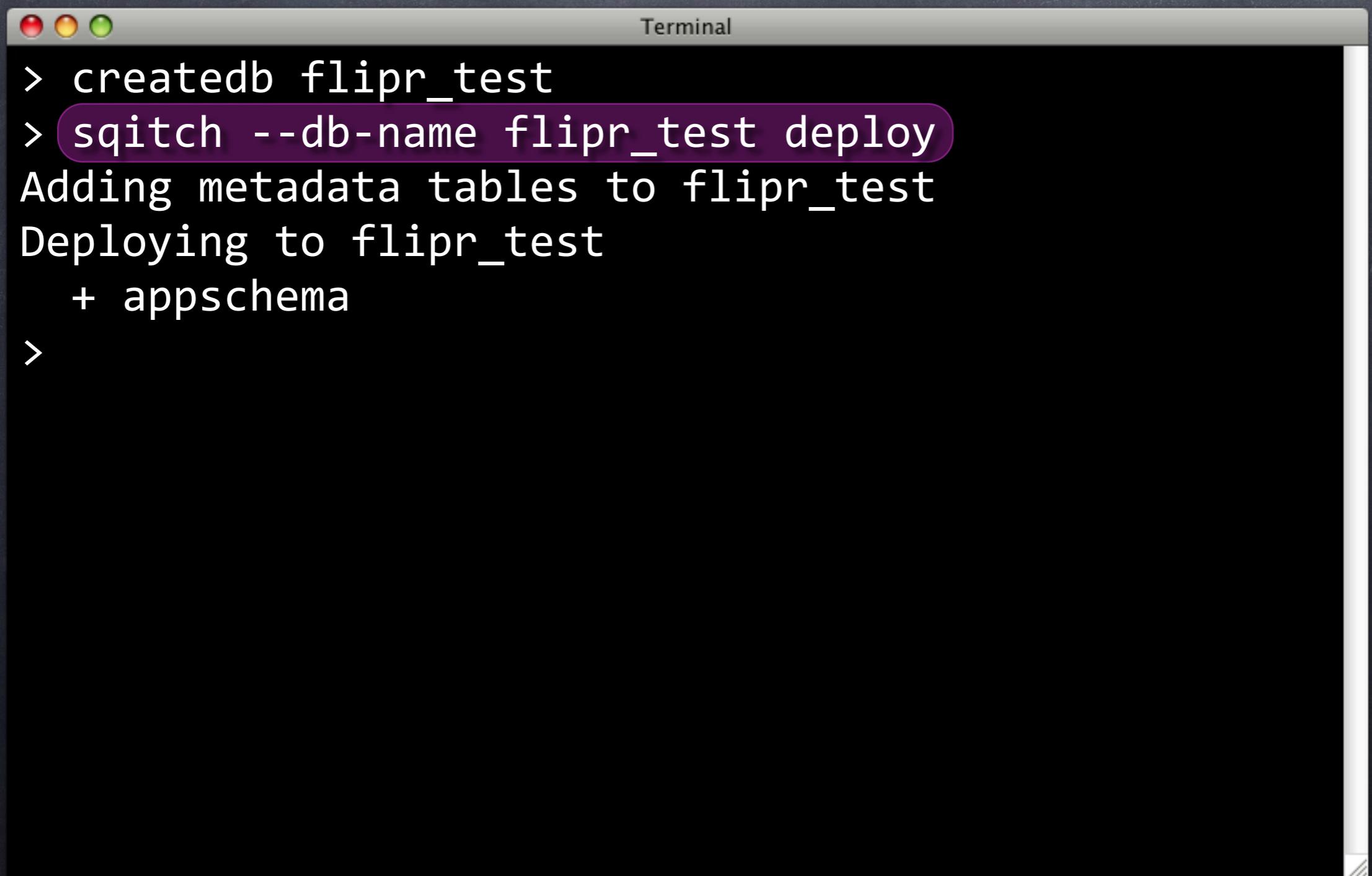


A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the following command-line session:

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

The first command, "createdb flipr_test", is highlighted with a green rounded rectangle.

Make it So!

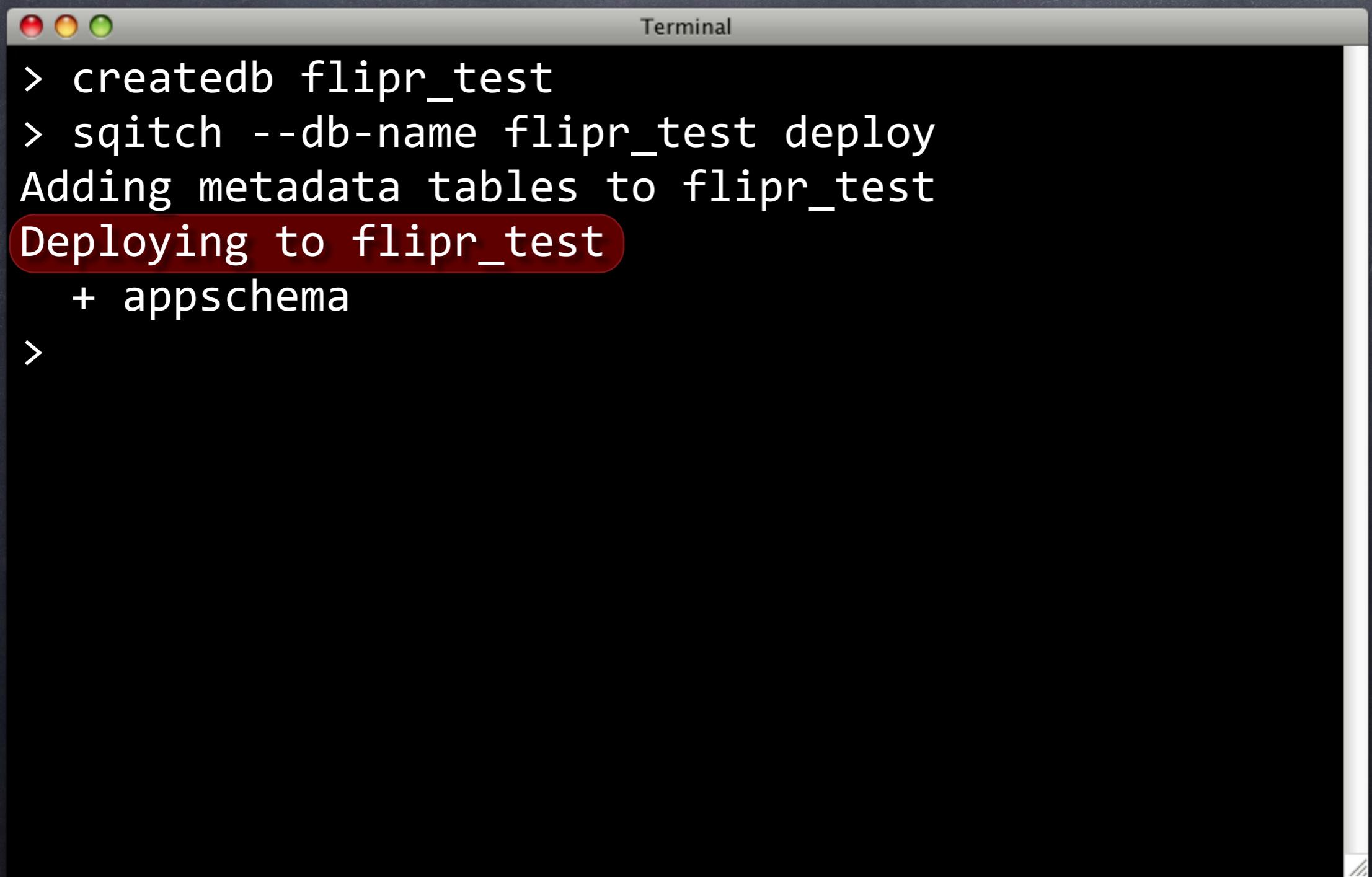


A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the following text:

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

The command "sqitch --db-name flipr_test deploy" is highlighted with a purple rounded rectangle.

Make it So!

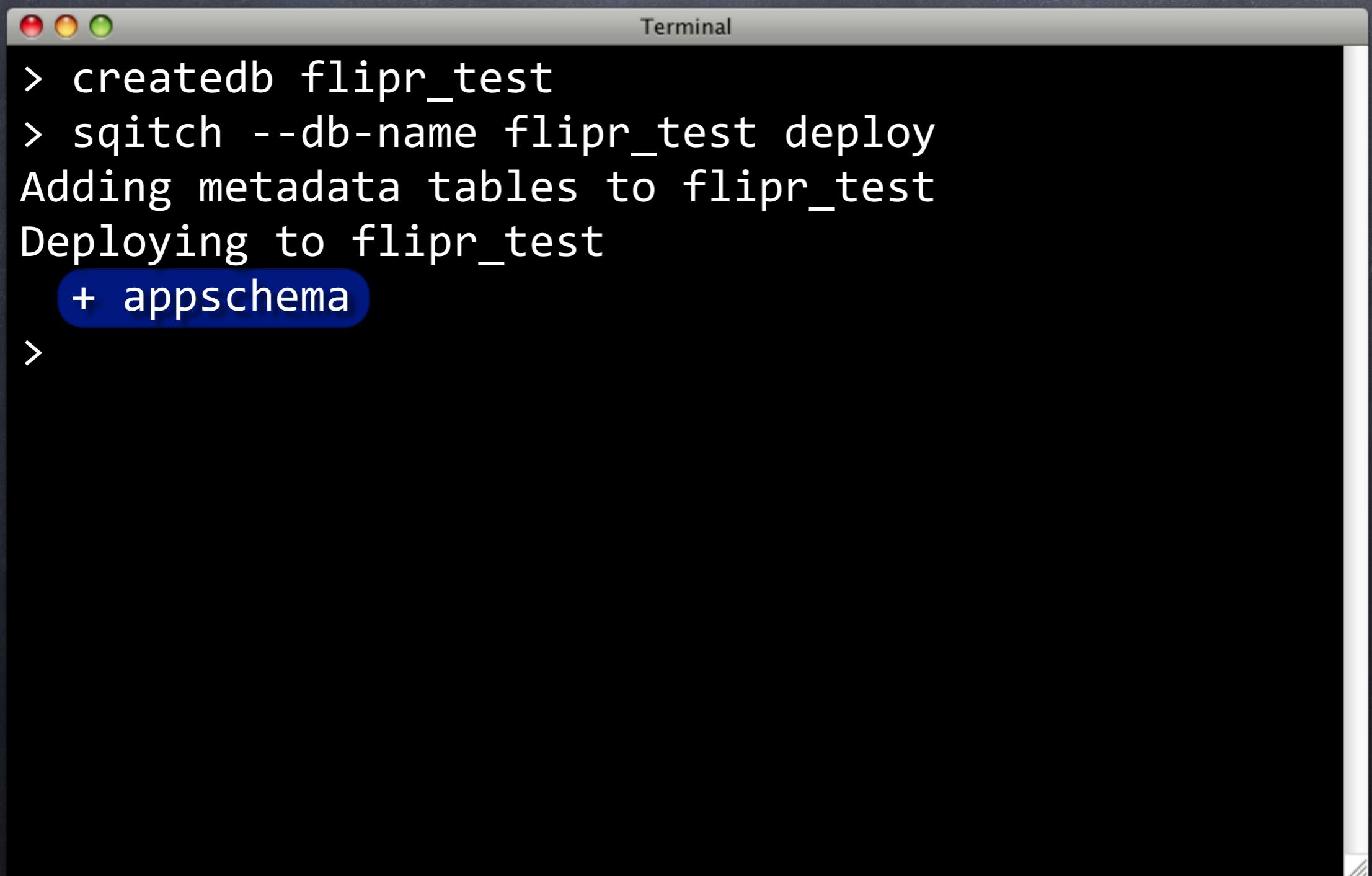


A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the following text:

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

The line "Deploying to flipr_test" is highlighted with a red rounded rectangle.

Make it So!



A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the following command-line session:

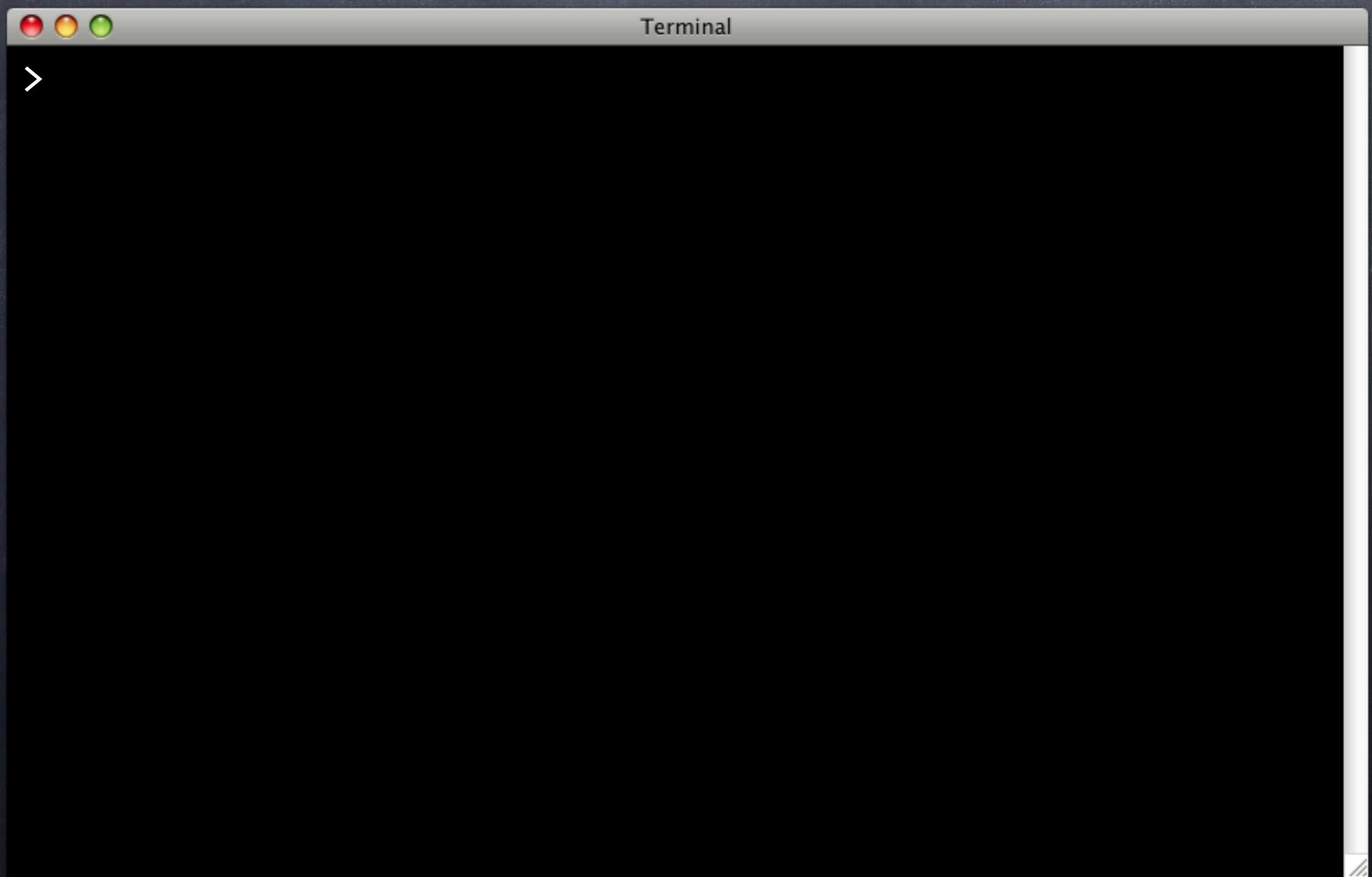
```
> createdb flipr_test
> sqitch --db-name flipr_test deploy
Adding metadata tables to flipr_test
Deploying to flipr_test
+ appschema
>
```

The word "+ appschema" is highlighted with a blue rounded rectangle.

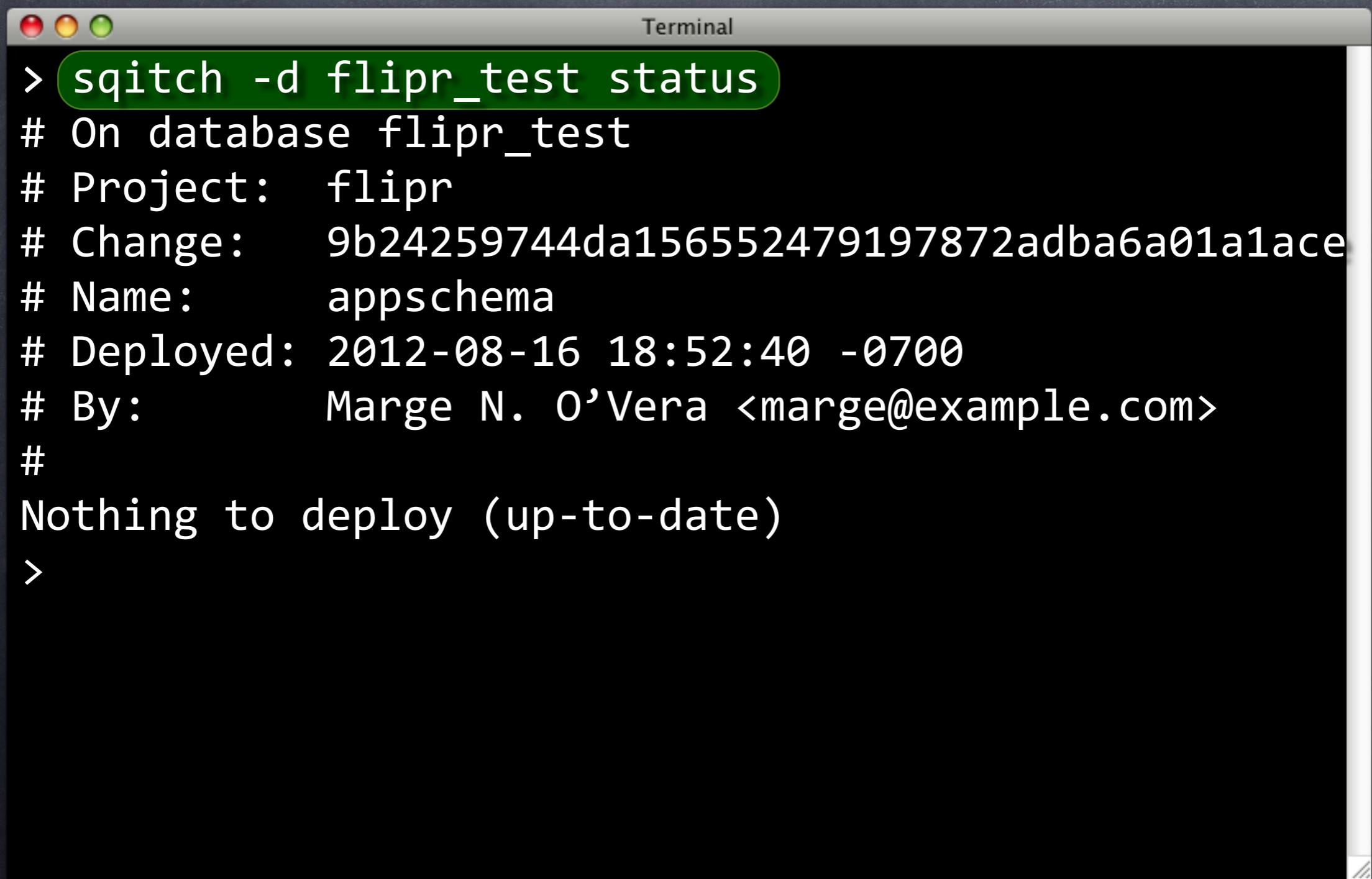
Make it So!

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

How's it Look?



How's it Look?



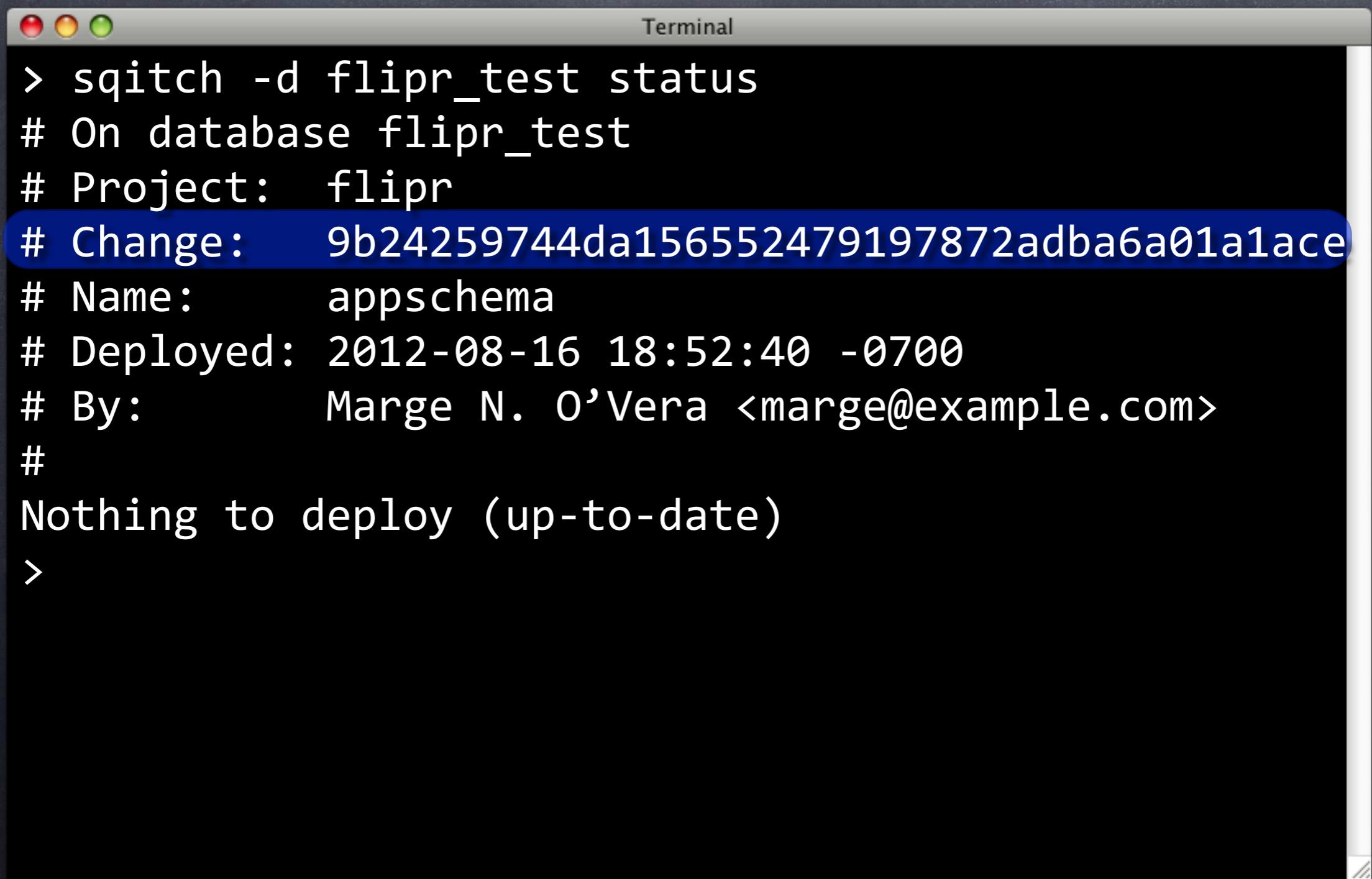
A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the following text:

```
> sqitch -d flipr_test status
# On database flipr_test
# Project: flipr
# Change: 9b24259744da156552479197872adba6a01a1ace
# Name: appschema
# Deployed: 2012-08-16 18:52:40 -0700
# By: Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

How's it Look?

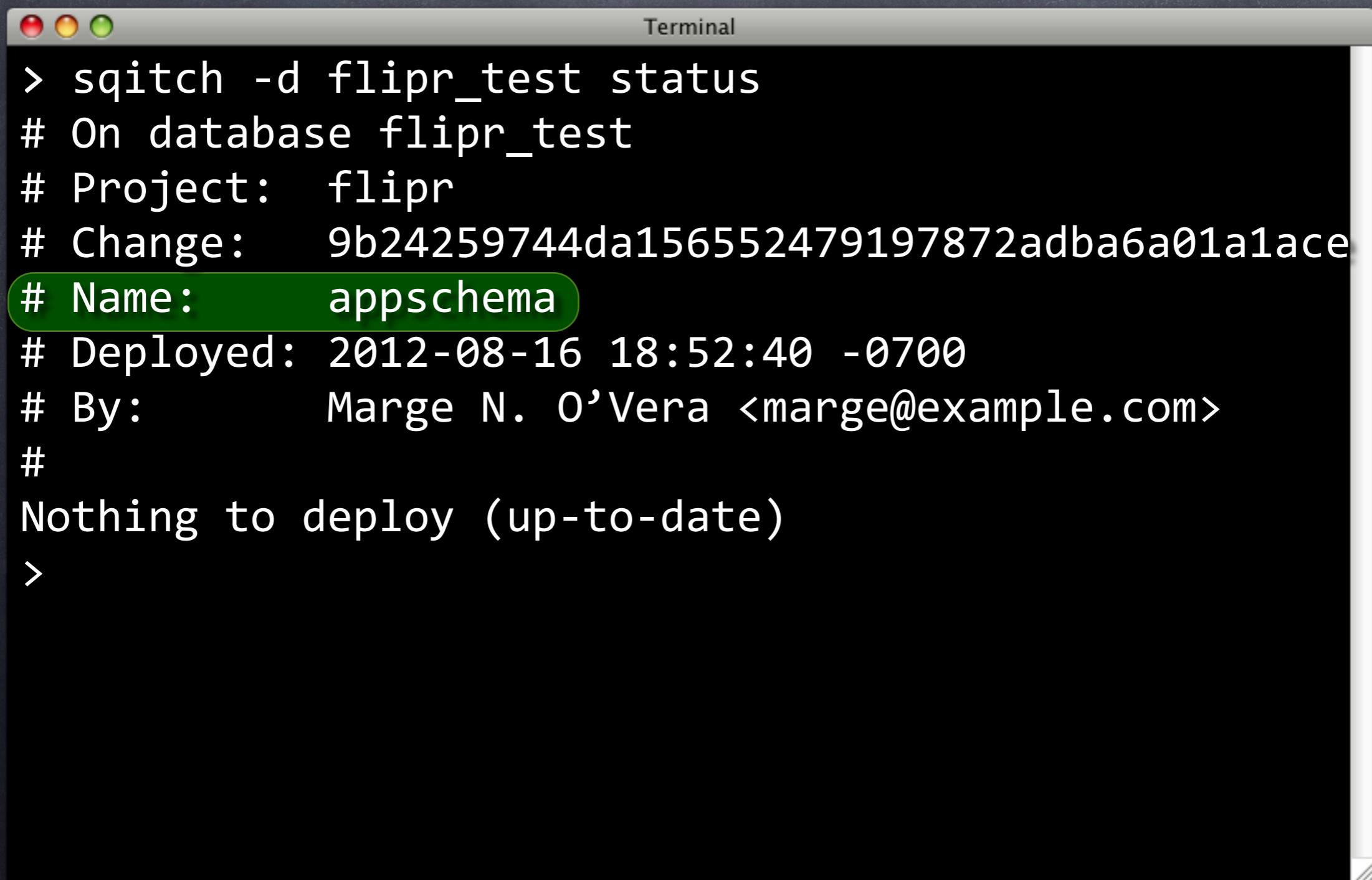
```
Terminal  
> sqitch -d flipr_test status  
# On database flipr_test  
# Project: flipr  
# Change: 9b24259744da156552479197872adba6a01a1ace  
# Name: appschema  
# Deployed: 2012-08-16 18:52:40 -0700  
# By: Marge N. O'Vera <marge@example.com>  
#  
Nothing to deploy (up-to-date)  
>
```

How's it Look?

A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the output of a "sqitch -d flipr_test status" command. The output shows a single change (9b24259744da156552479197872adb6a01a1ace) with details: Name: appschema, Deployed: 2012-08-16 18:52:40 -0700, By: Marge N. O'Vera <marge@example.com>. A blue rectangular highlight covers the "Change:" line and its associated details. The text "Nothing to deploy (up-to-date)" is also present.

```
> sqitch -d flipr_test status
# On database flipr_test
# Project: flipr
# Change: 9b24259744da156552479197872adb6a01a1ace
#   Name:      appschema
#   Deployed: 2012-08-16 18:52:40 -0700
#   By:        Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

How's it Look?



A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the following text output from the command "sqitch -d flipr_test status":

```
> sqitch -d flipr_test status
# On database flipr_test
# Project: flipr
# Change: 9b24259744da156552479197872adba6a01a1ace
# Name: appschema
# Deployed: 2012-08-16 18:52:40 -0700
# By: Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

The line "# Name: appschema" is highlighted with a green rounded rectangle.

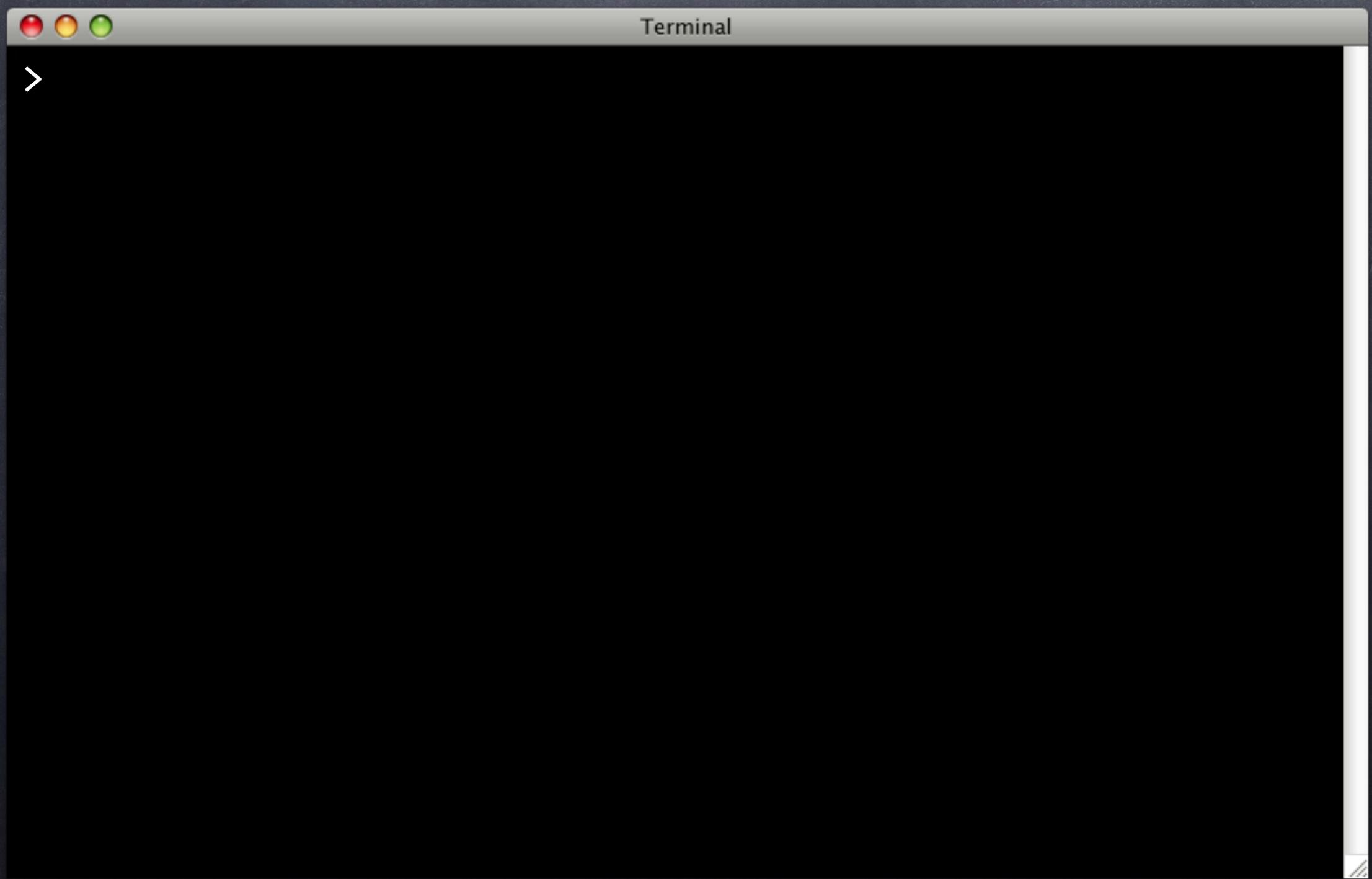
How's it Look?

```
Terminal  
> sqitch -d flipr_test status  
# On database flipr_test  
# Project: flipr  
# Change: 9b24259744da156552479197872adba6a01a1ace  
# Name: appschema  
# Deployed: 2012-08-16 18:52:40 -0700  
# By: Marge N. O'Vera <marge@example.com>  
#  
Nothing to deploy (up-to-date)  
>
```

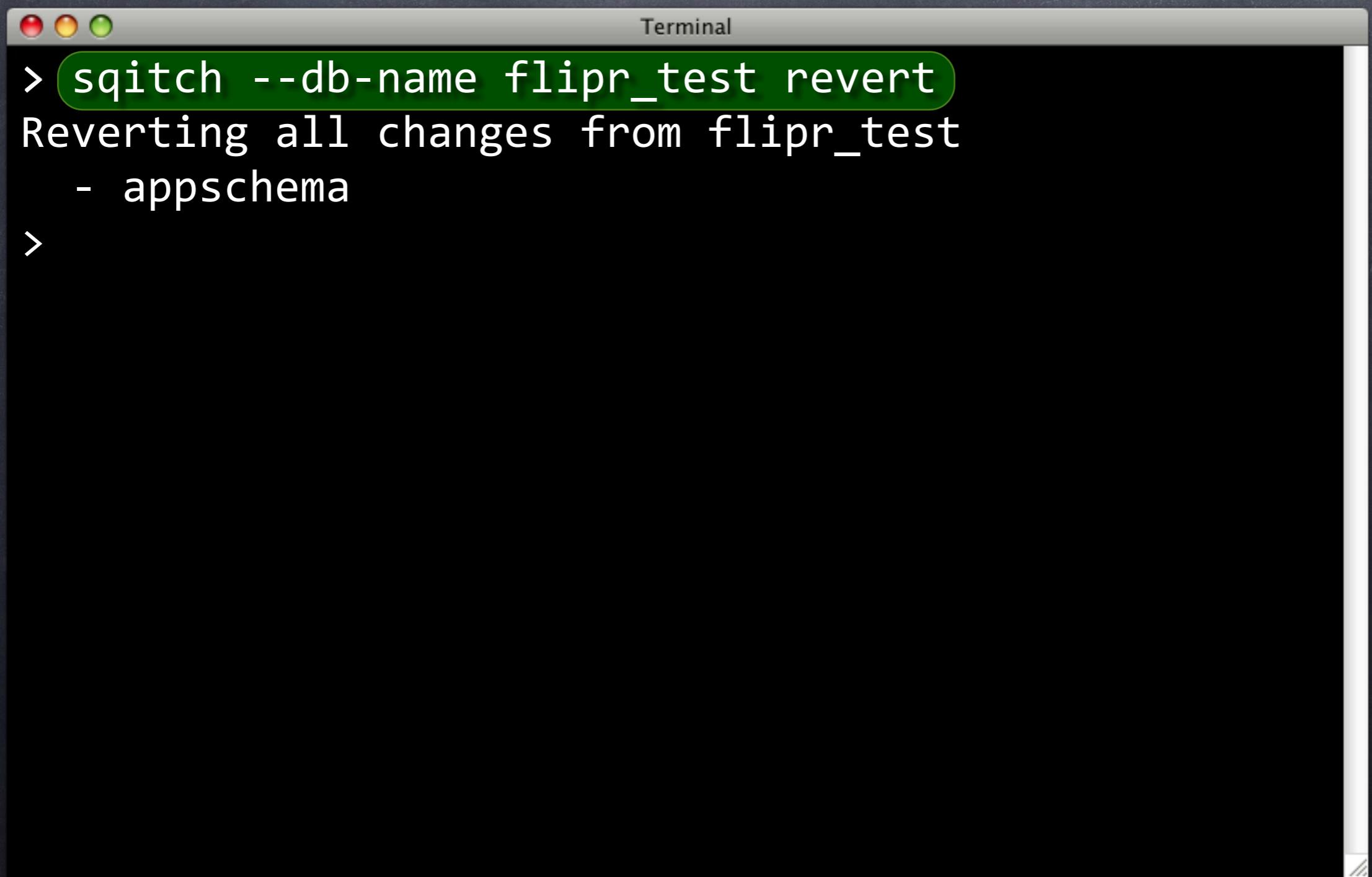
How's it Look?

```
Terminal  
> sqitch -d flipr_test status  
# On database flipr_test  
# Project: flipr  
# Change: 9b24259744da156552479197872adba6a01a1ace  
# Name: appschema  
# Deployed: 2012-08-16 18:52:40 -0700  
# By: Marge N. O'Vera <marge@example.com>  
#  
Nothing to deploy (up-to-date)  
>
```

Go Back



Go Back

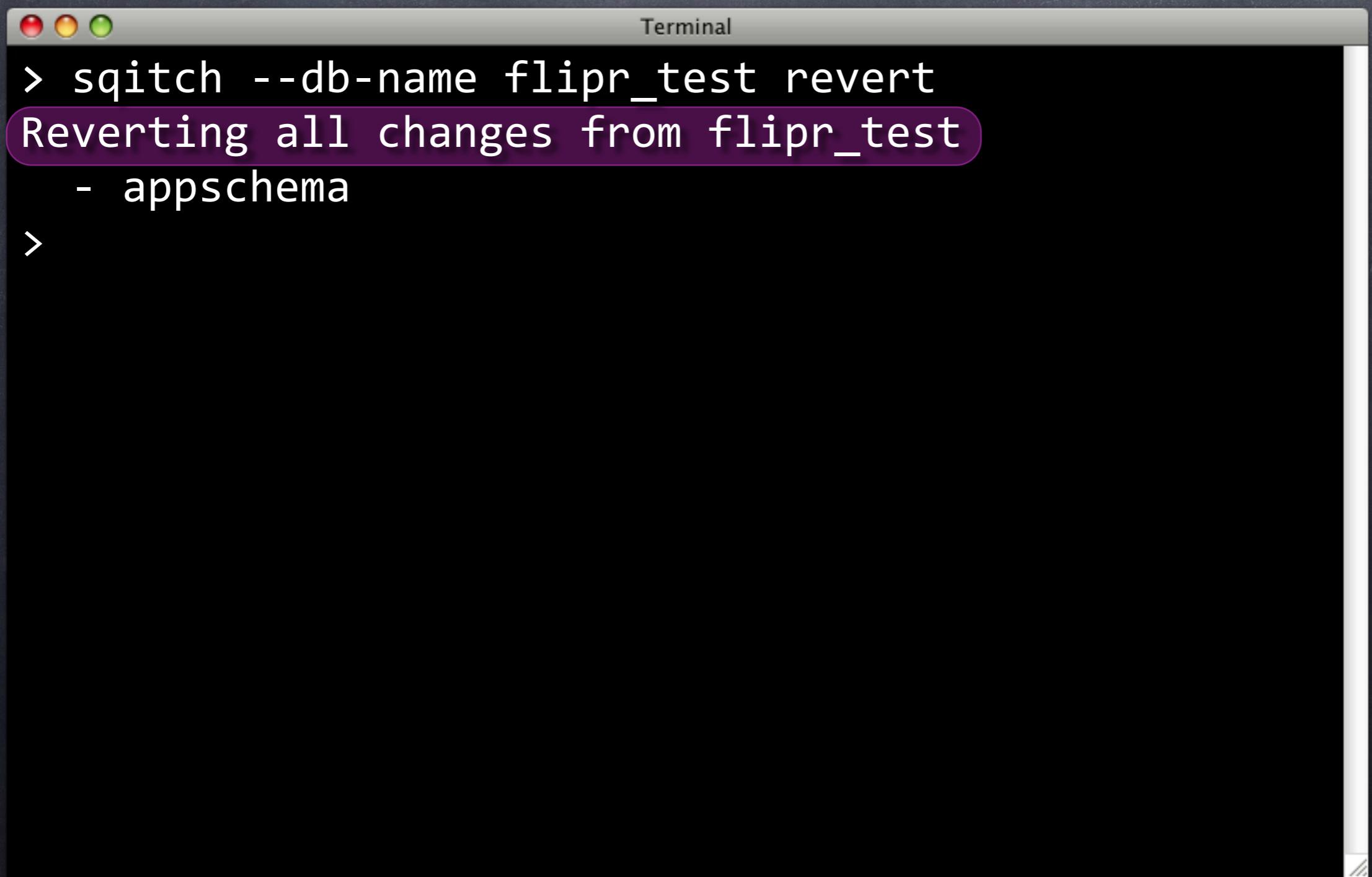


A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The title bar says "Terminal". The main pane contains the following text:

```
> sqitch --db-name flipr_test revert
Reverting all changes from flipr_test
- appschema
>
```

The command "sqitch --db-name flipr_test revert" is highlighted with a green rounded rectangle.

Go Back

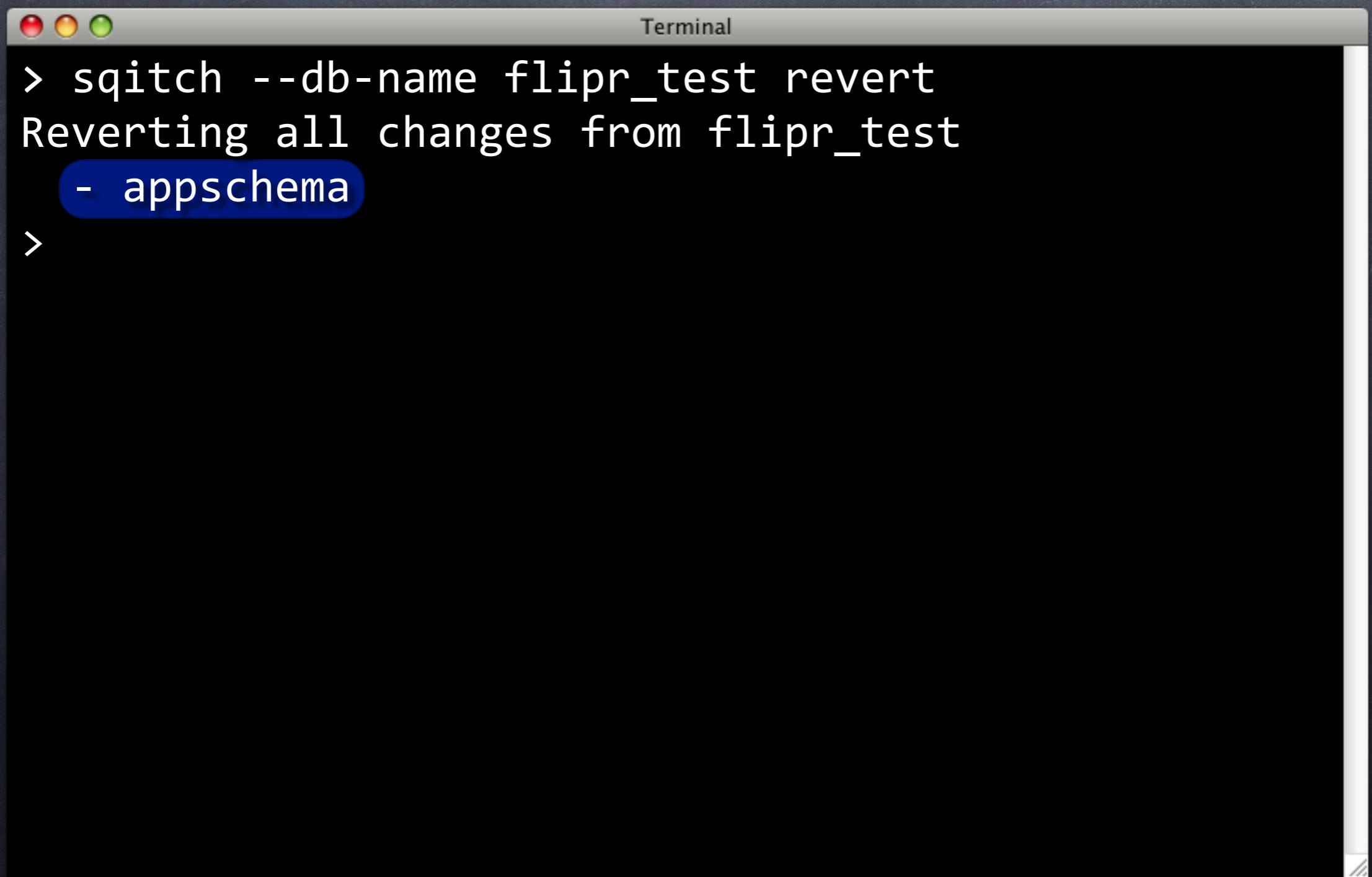


A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The title bar says "Terminal". The main pane contains the following text:

```
> sqitch --db-name flipr_test revert
Reverting all changes from flipr_test
- appschema
>
```

The text "Reverting all changes from flipr_test" is highlighted with a purple rounded rectangle.

Go Back

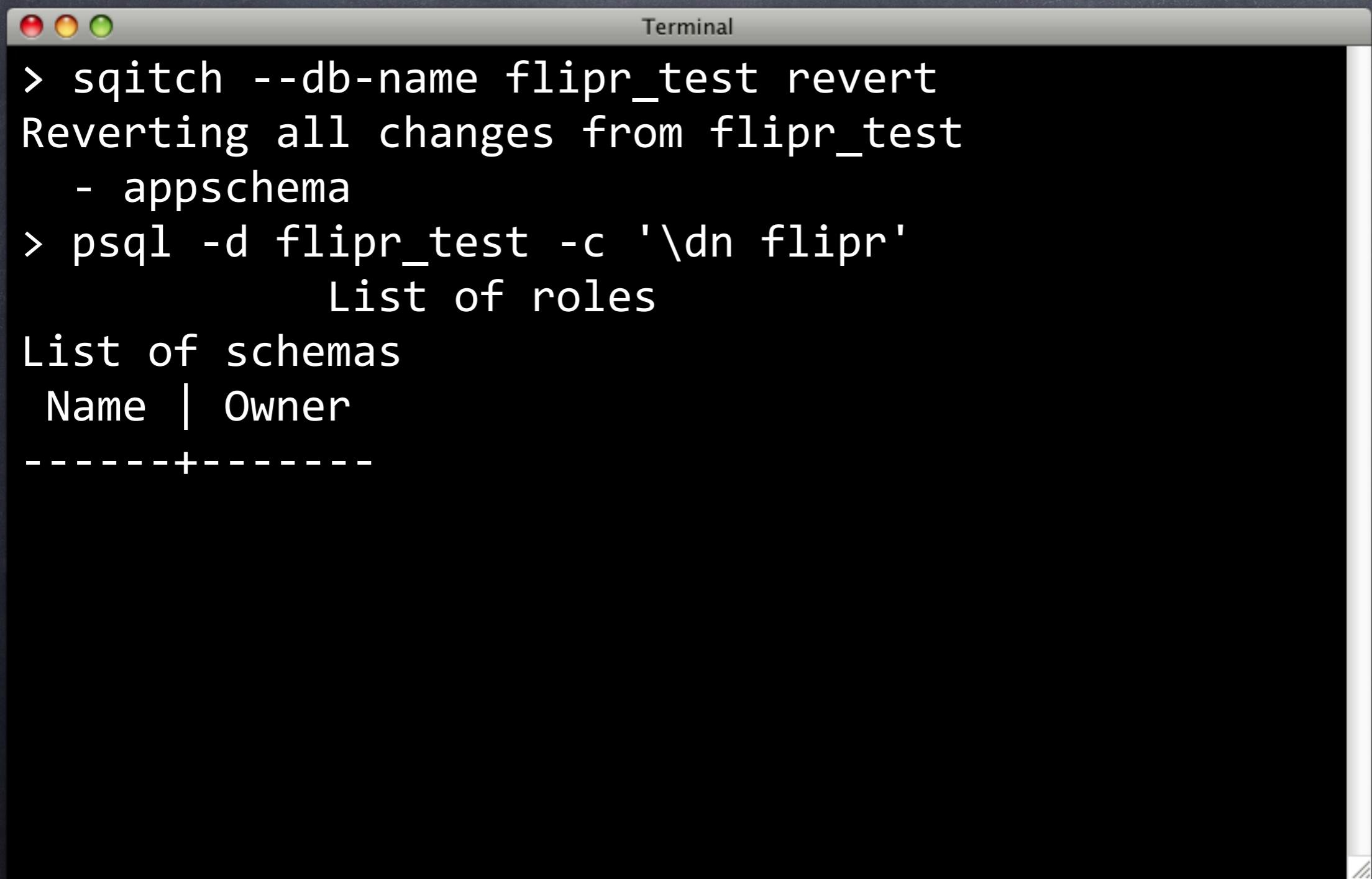


A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The text area contains the following command and its execution:

```
> sqitch --db-name flipr_test revert
Reverting all changes from flipr_test
  - appschema
>
```

The word "appschema" is highlighted with a blue rounded rectangle.

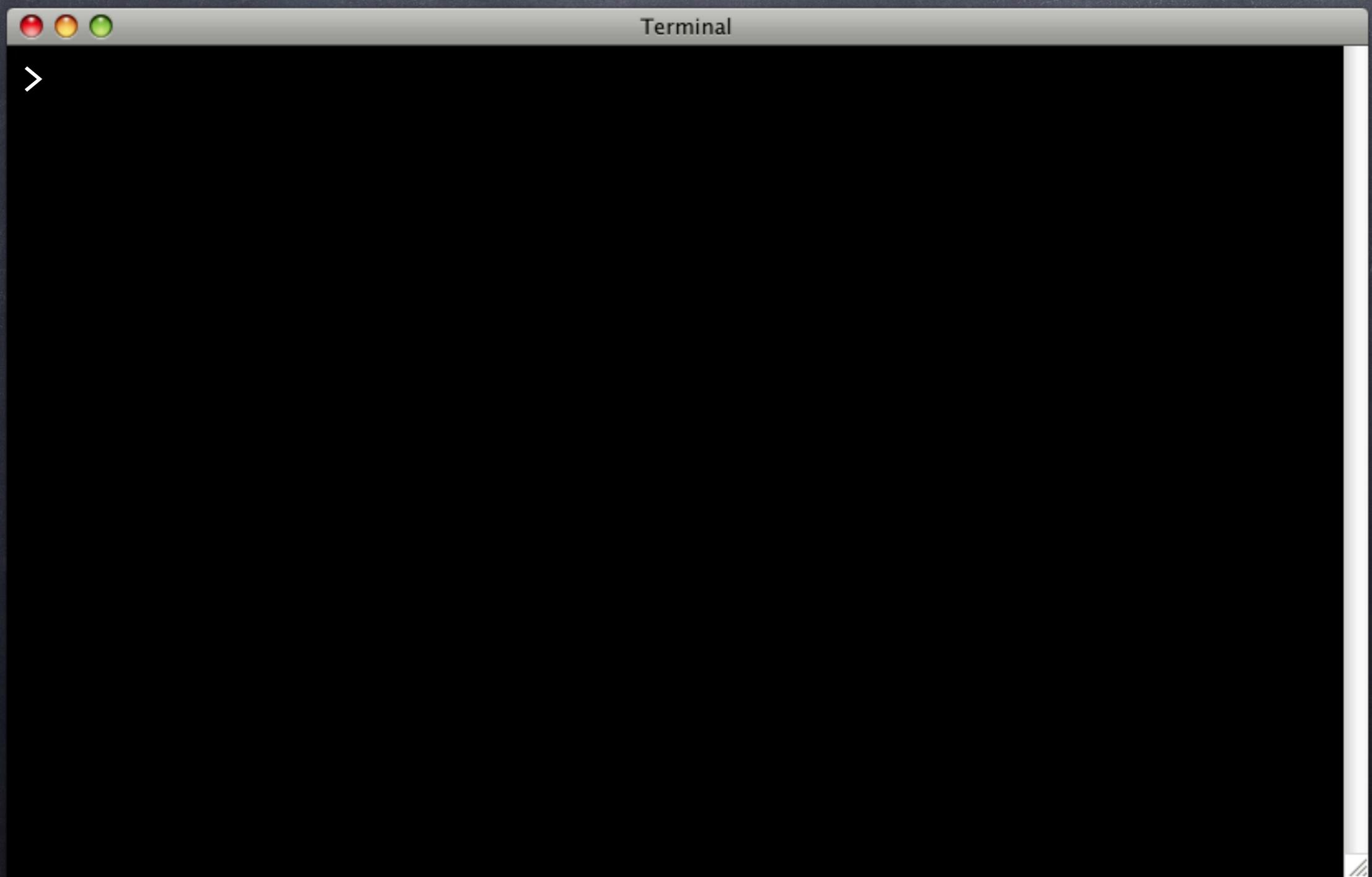
Go Back



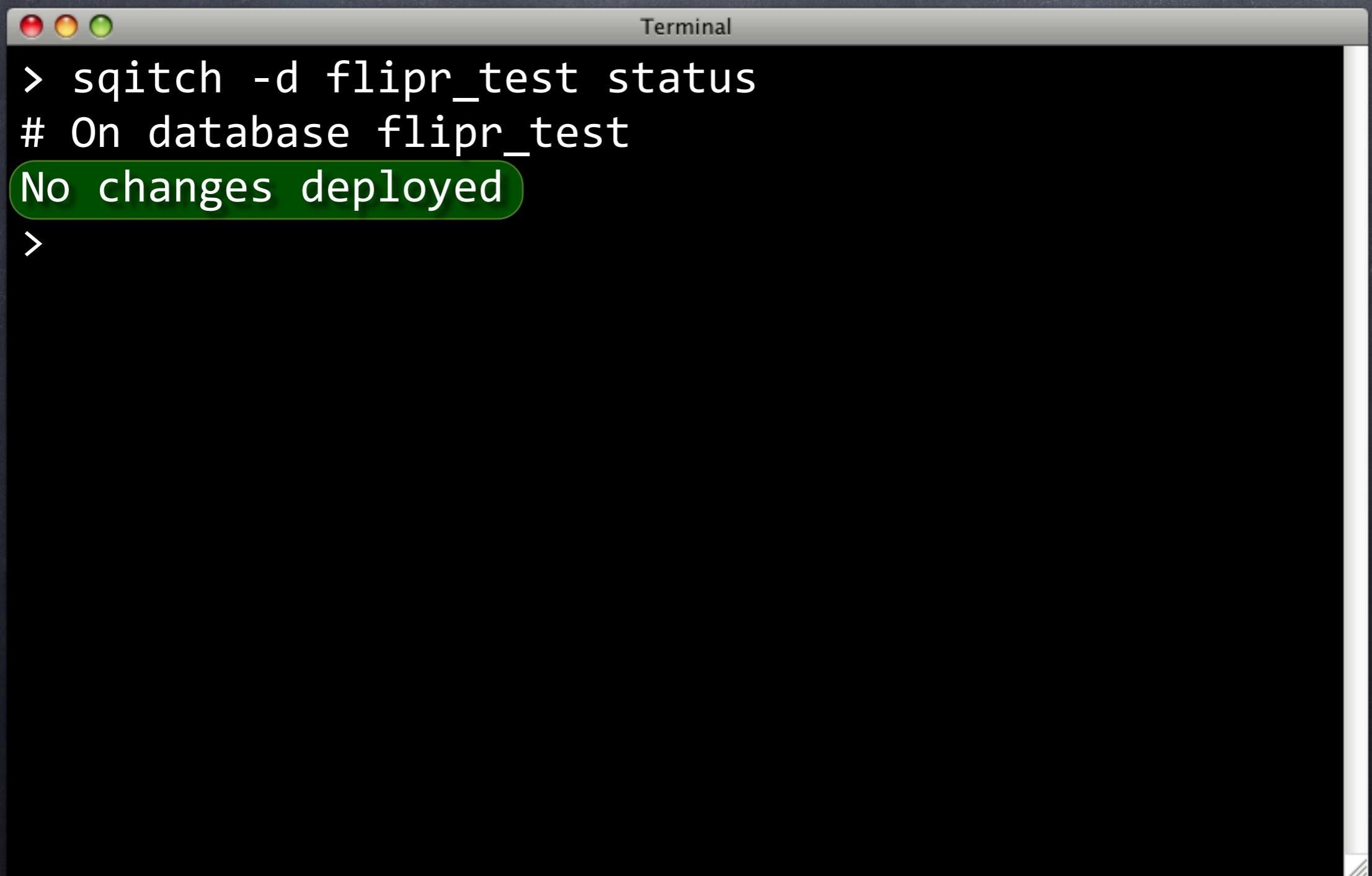
A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the following command-line session:

```
> sqitch --db-name flipr_test revert
Reverting all changes from flipr_test
- appschema
> psql -d flipr_test -c '\dn flipr'
      List of roles
List of schemas
  Name | Owner
-----+-----
```

What's The Status?

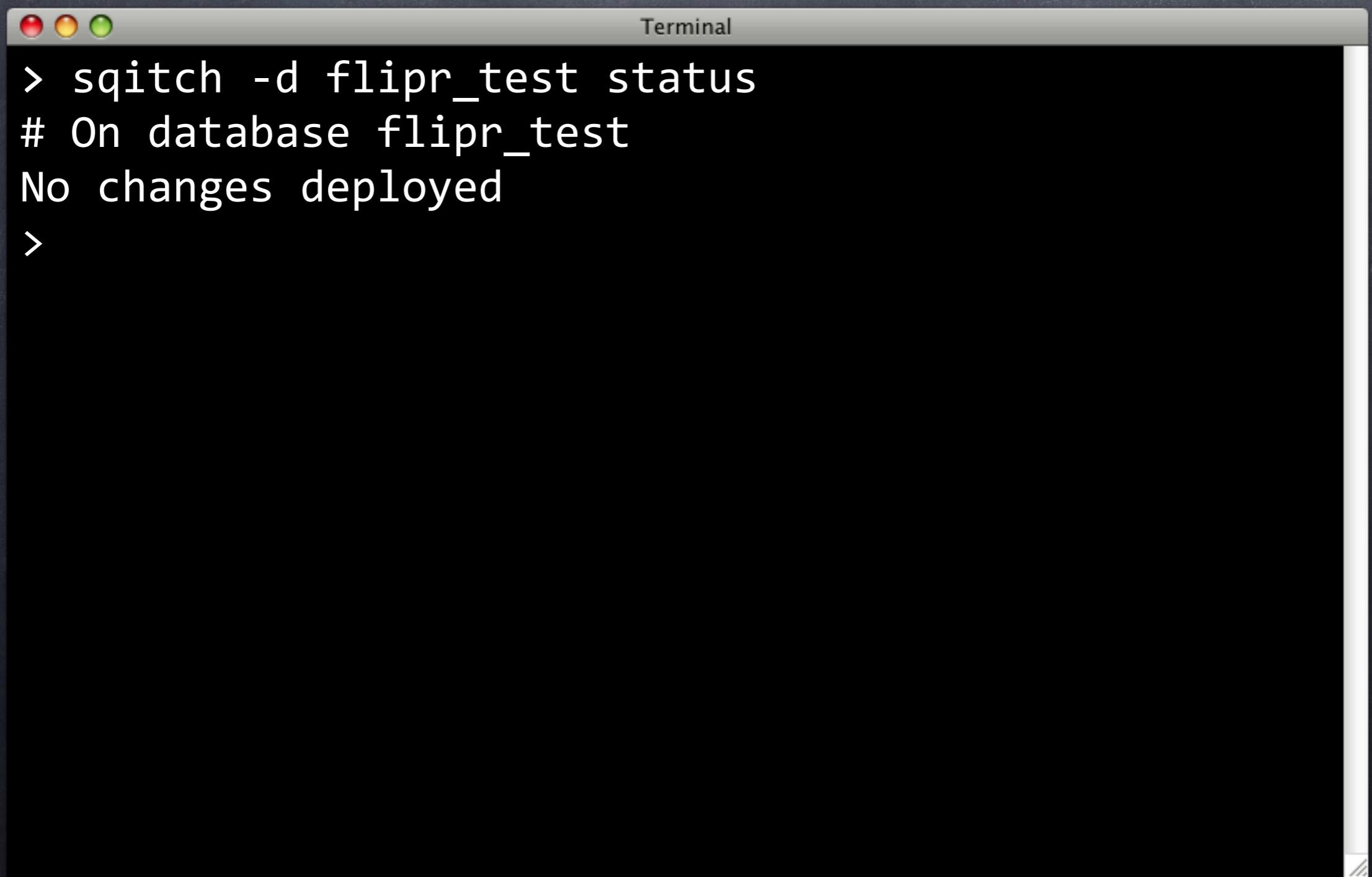


What's The Status?



```
Terminal  
> sqitch -d flipr_test status  
# On database flipr_test  
No changes deployed  
>
```

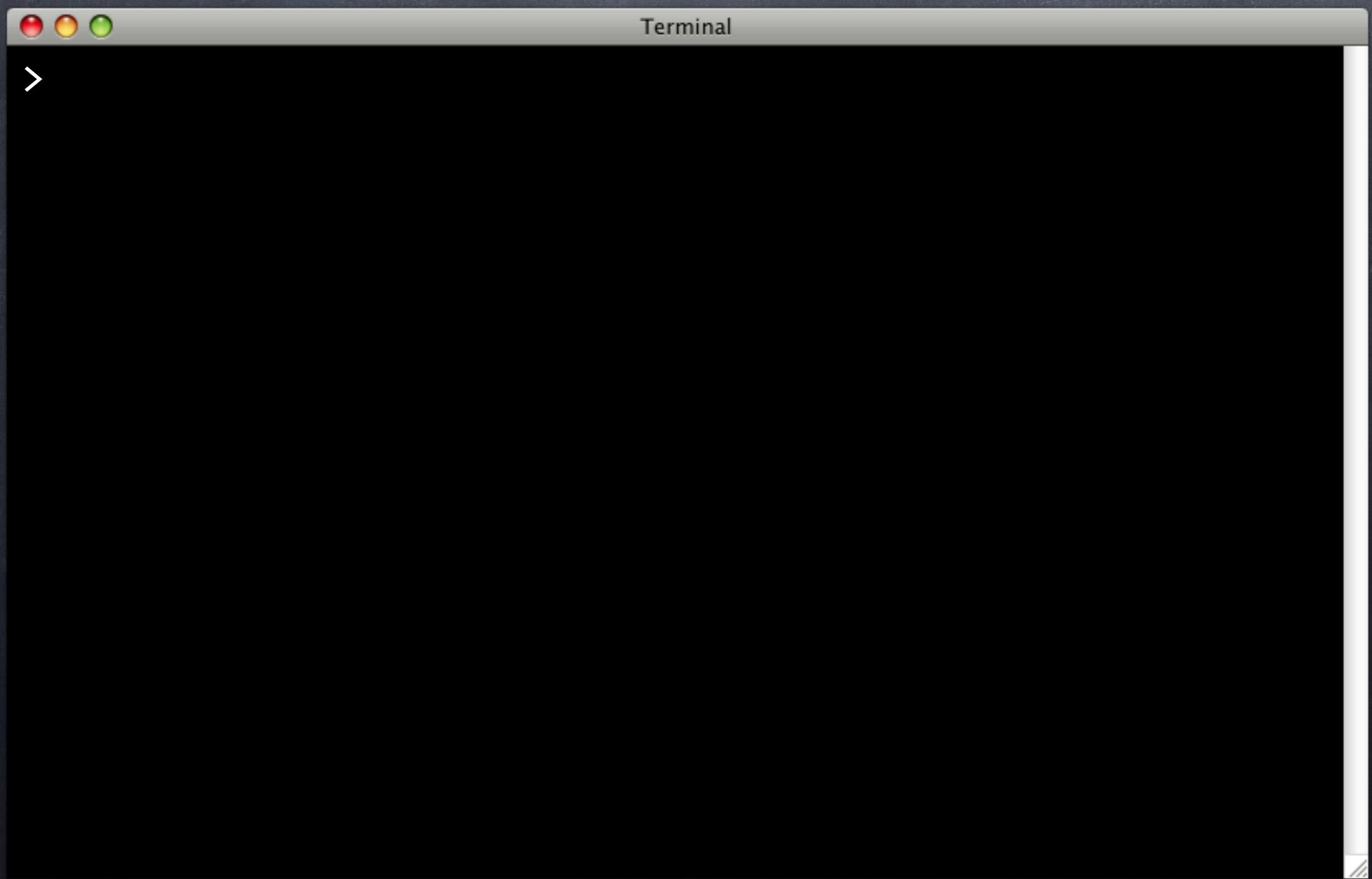
What's The Status?

A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The title bar reads "Terminal". The main pane contains the following text:

```
> sqitch -d flipr_test status
# On database flipr_test
No changes deployed
>
```

The terminal window is set against a dark background.

History



History

```
Terminal  
> sqitch -d flipr_test log  
On database flipr_test  
Revert 9b24259744da156552479197872adba6a01a1ace  
Name: appschema  
Committer: Marge N. O'Vera <marge@example.com>  
Date: 2012-08-16 18:54:55 -0700  
  
App schema for all flipr objects.  
  
Deploy 9b24259744da156552479197872adba6a01a1ace  
Name: appschema  
Committer: Marge N. O'Vera <marge@example.com>  
Date: 2012-08-16 18:52:40 -0700  
  
App schema for all flipr objects.
```

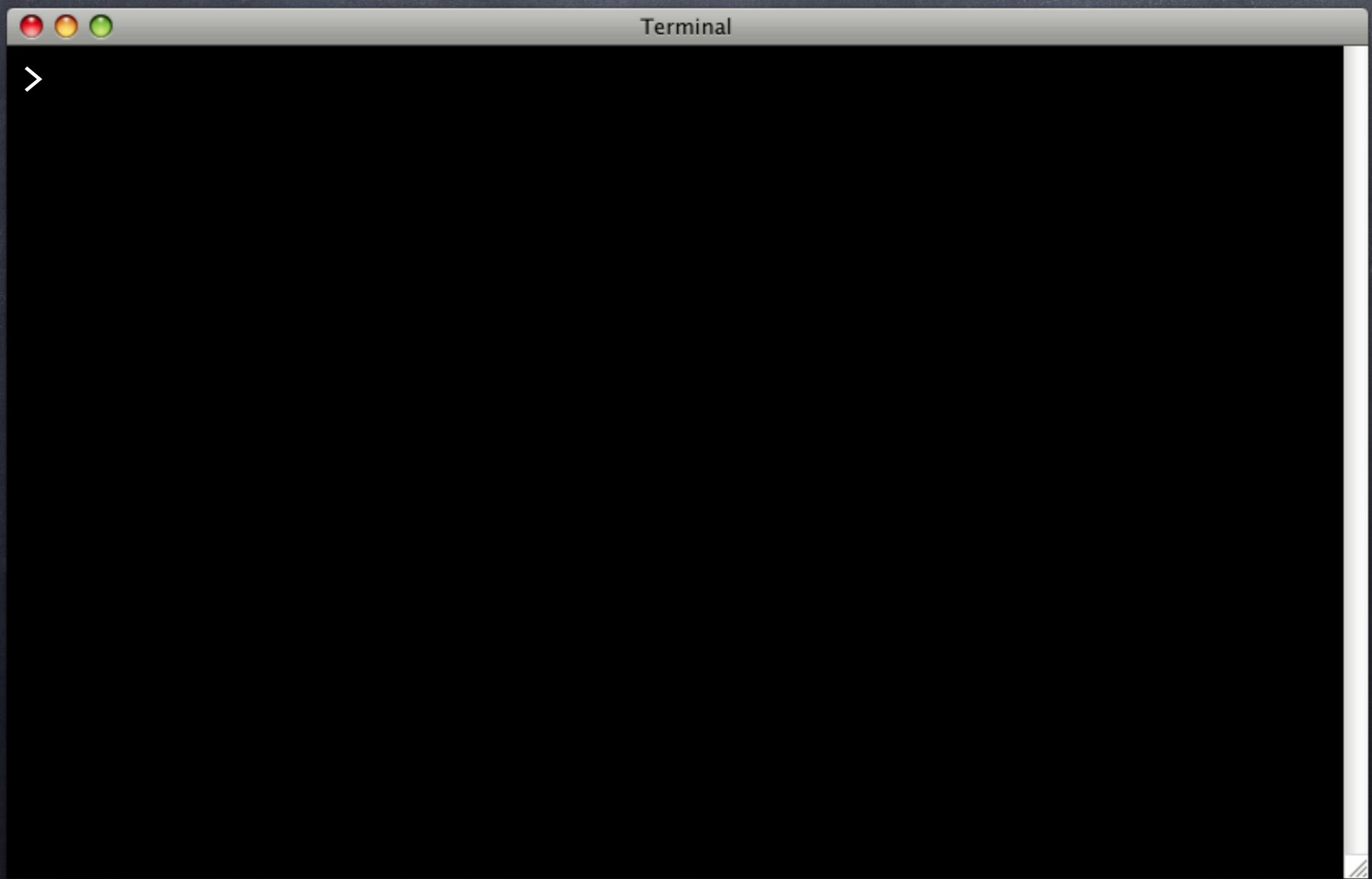
History

```
Terminal  
> sqitch -d flipr_test log  
On database flipr_test  
Revert 9b24259744da156552479197872adba6a01a1ace  
Name:      appschema  
Committer: Marge N. O'Vera <marge@example.com>  
Date:      2012-08-16 18:54:55 -0700  
  
App schema for all flipr objects.  
  
Deploy 9b24259744da156552479197872adba6a01a1ace  
Name:      appschema  
Committer: Marge N. O'Vera <marge@example.com>  
Date:      2012-08-16 18:52:40 -0700  
  
App schema for all flipr objects.
```

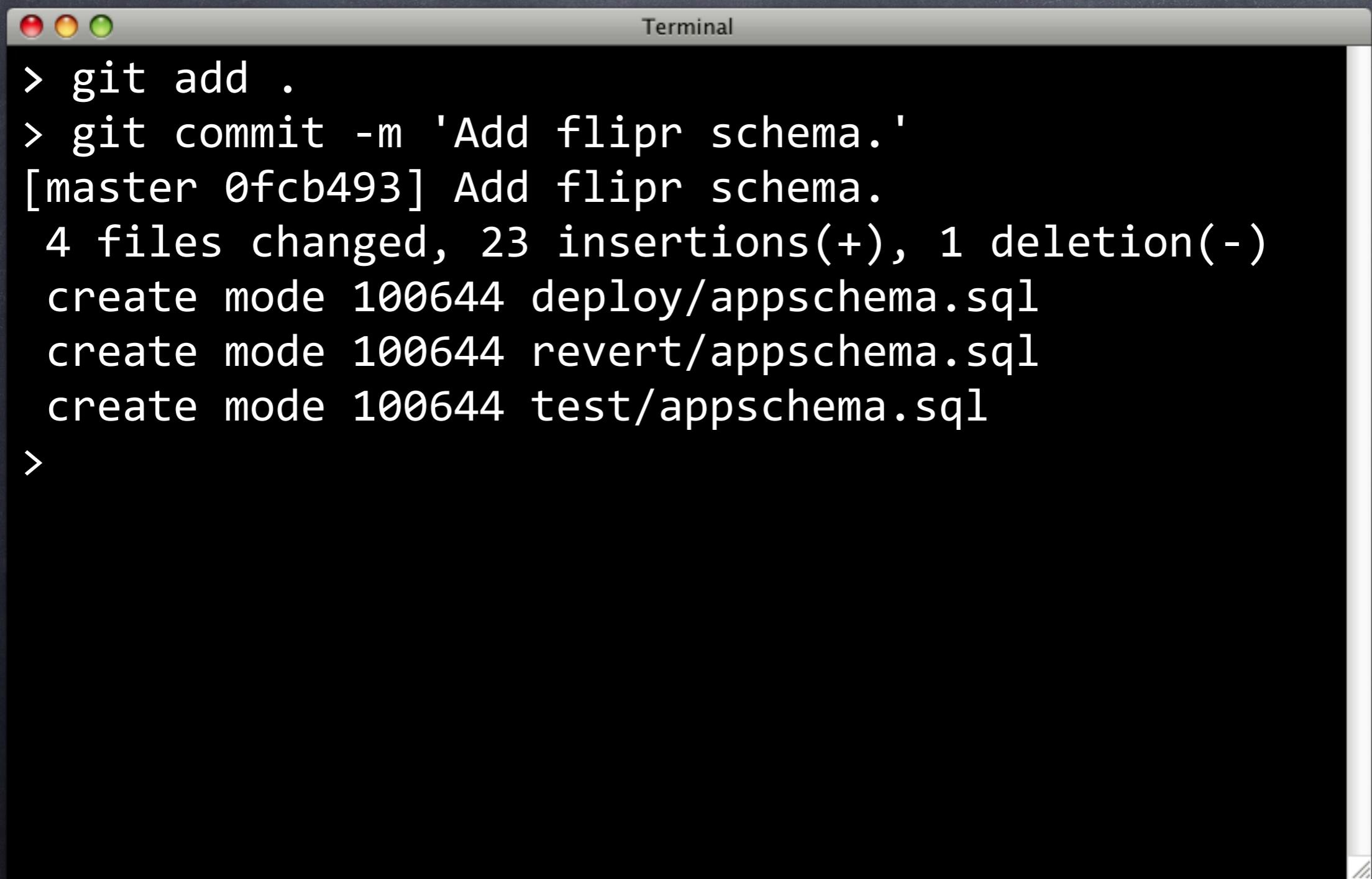
History

```
Terminal  
> sqitch -d flipr_test log  
On database flipr_test  
Revert 9b24259744da156552479197872adba6a01a1ace  
Name:      appschema  
Committer: Marge N. O'Vera <marge@example.com>  
Date:      2012-08-16 18:54:55 -0700  
  
App schema for all flipr objects.  
  
Deploy 9b24259744da156552479197872adba6a01a1ace  
Name:      appschema  
Committer: Marge N. O'Vera <marge@example.com>  
Date:      2012-08-16 18:52:40 -0700  
  
App schema for all flipr objects.
```

Commit It!



Commit It!

A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The terminal itself is black with white text. It shows a series of git commands being run:

```
> git add .
> git commit -m 'Add flipr schema.'
[master 0fcf493] Add flipr schema.
 4 files changed, 23 insertions(+), 1 deletion(-)
 create mode 100644 deploy/appschema.sql
 create mode 100644 revert/appschema.sql
 create mode 100644 test/appschema.sql
>
```

The commit message "Add flipr schema." is enclosed in single quotes. The commit output shows the creation of three new files: "deploy/appschema.sql", "revert/appschema.sql", and "test/appschema.sql", each with mode 100644.

Commit It!

```
Terminal  
> git add .  
> git commit -m 'Add flipr schema.'  
[master 0fcf493] Add flipr schema.  
 4 files changed, 23 insertions(+), 1 deletion(-)  
create mode 100644 deploy/appschema.sql  
create mode 100644 revert/appschema.sql  
create mode 100644 test/appschema.sql  
> sqitch --db-name flipr_test deploy  
Deploying changes to flipr_test  
  + appschema  
>
```

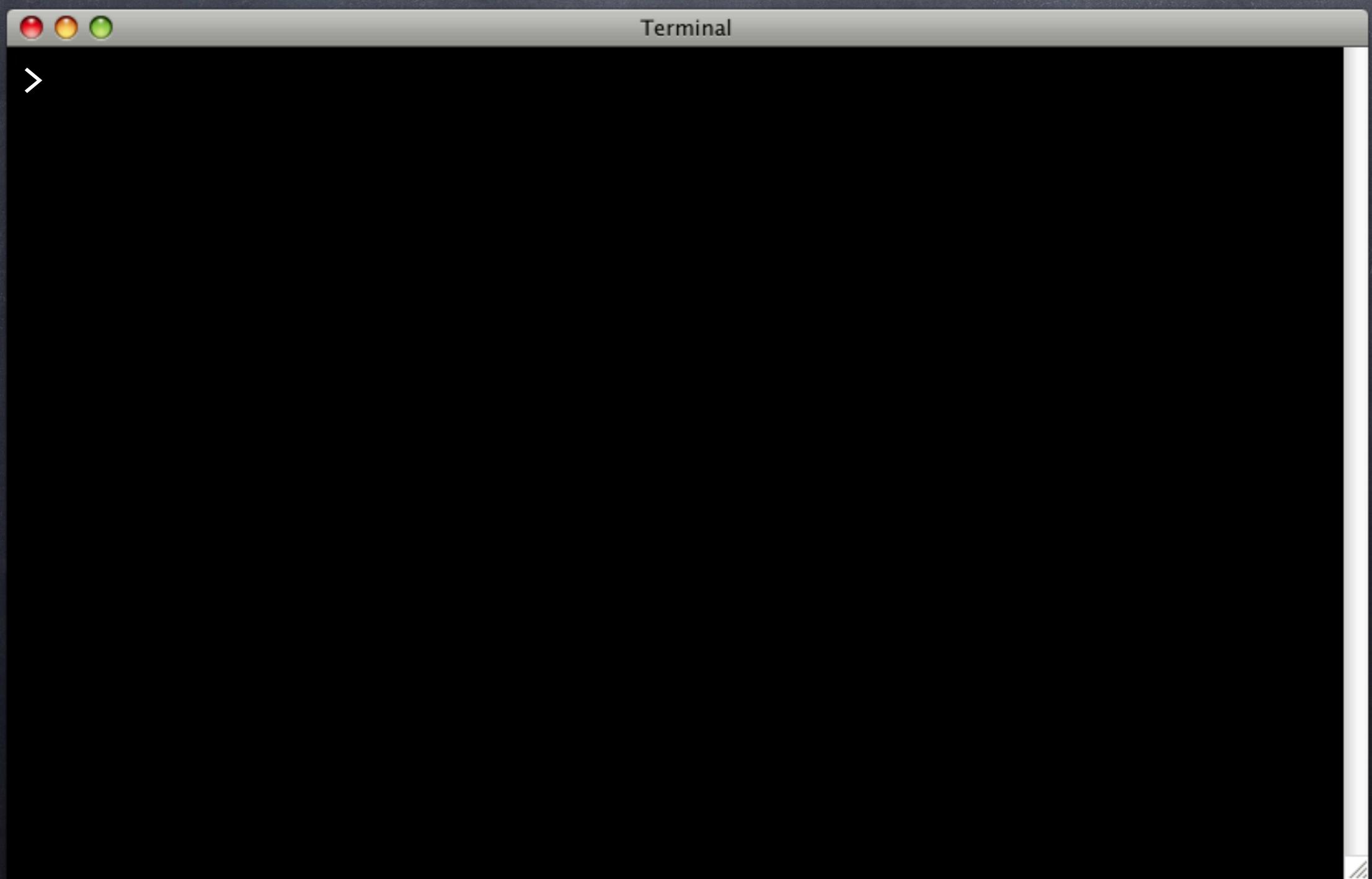
Commit It!

```
Terminal  
> git add .  
> git commit -m 'Add flipr schema.'  
[master 0fcf493] Add flipr schema.  
 4 files changed, 23 insertions(+), 1 deletion(-)  
create mode 100644 deploy/appschema.sql  
create mode 100644 revert/appschema.sql  
create mode 100644 test/appschema.sql  
> sqitch --db-name flipr_test deploy  
Deploying changes to flipr_test  
  + appschema  
>
```

Commit It!

```
Terminal  
> git add .  
> git commit -m 'Add flipr schema.'  
[master 0fcf493] Add flipr schema.  
 4 files changed, 23 insertions(+), 1 deletion(-)  
create mode 100644 deploy/appschema.sql  
create mode 100644 revert/appschema.sql  
create mode 100644 test/appschema.sql  
> sqitch --db-name flipr_test deploy  
Deploying changes to flipr_test  
  + appschema  
> psql -d flipr_test -c '\dn flipr'  
List of schemas  
 Name   | Owner  
-----+-----  
 flipr | marge
```

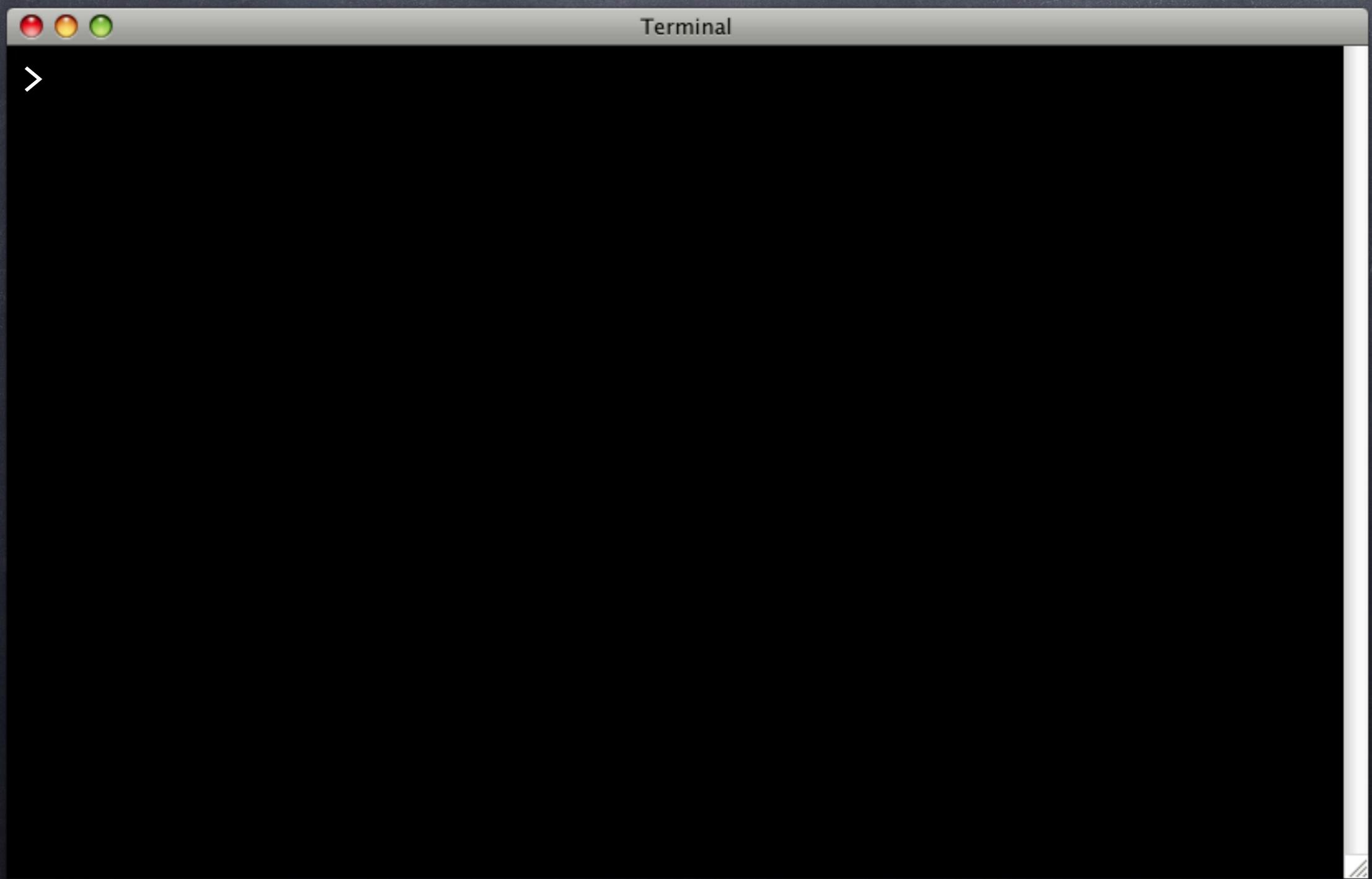
Status Update



Status Update

```
Terminal  
> sqitch -d flipr_test status  
# On database flipr_test  
# Project: flipr  
# Change: 9b24259744da156552479197872adb6a01a1ace  
# Name: appschema  
# Deployed: 2012-08-16 18:57:03 -0700  
# 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
>
```

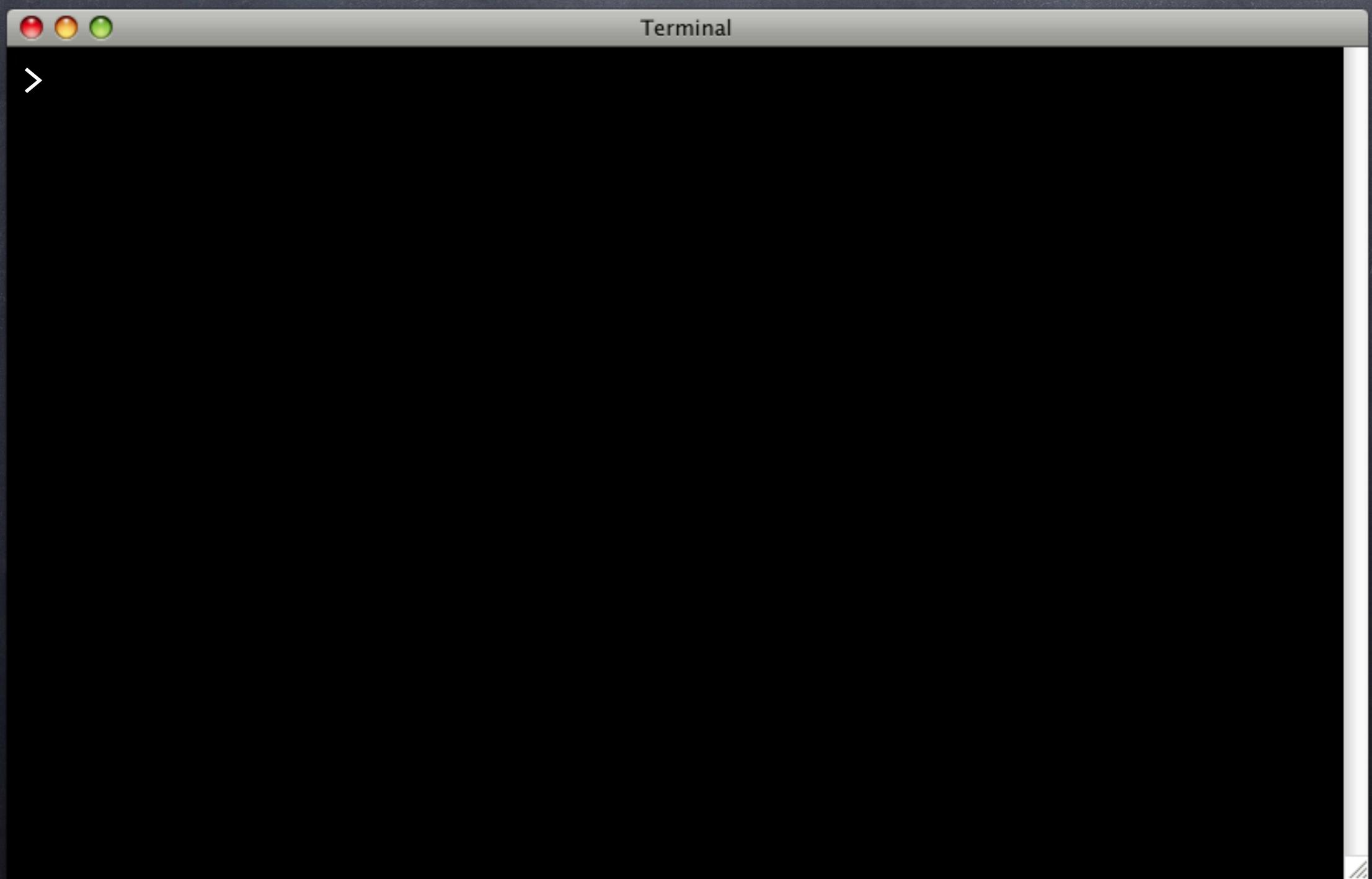
Save My Fingers

```
Terminal  
> sqitch config core.pg.db_name flipr_test  
> sqitch status  
# On database flipr_test  
# Project: flipr  
# Change: 9b24259744da1565524791  
# Name: appschema  
# Deployed: 2012-08-16 18:57:03 -0700  
# By: Marge N. O'Vera <marge@example.com>  
#  
Nothing to deploy (up-to-date)  
>
```



No --db-name

Dependencies!



Dependencies!

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

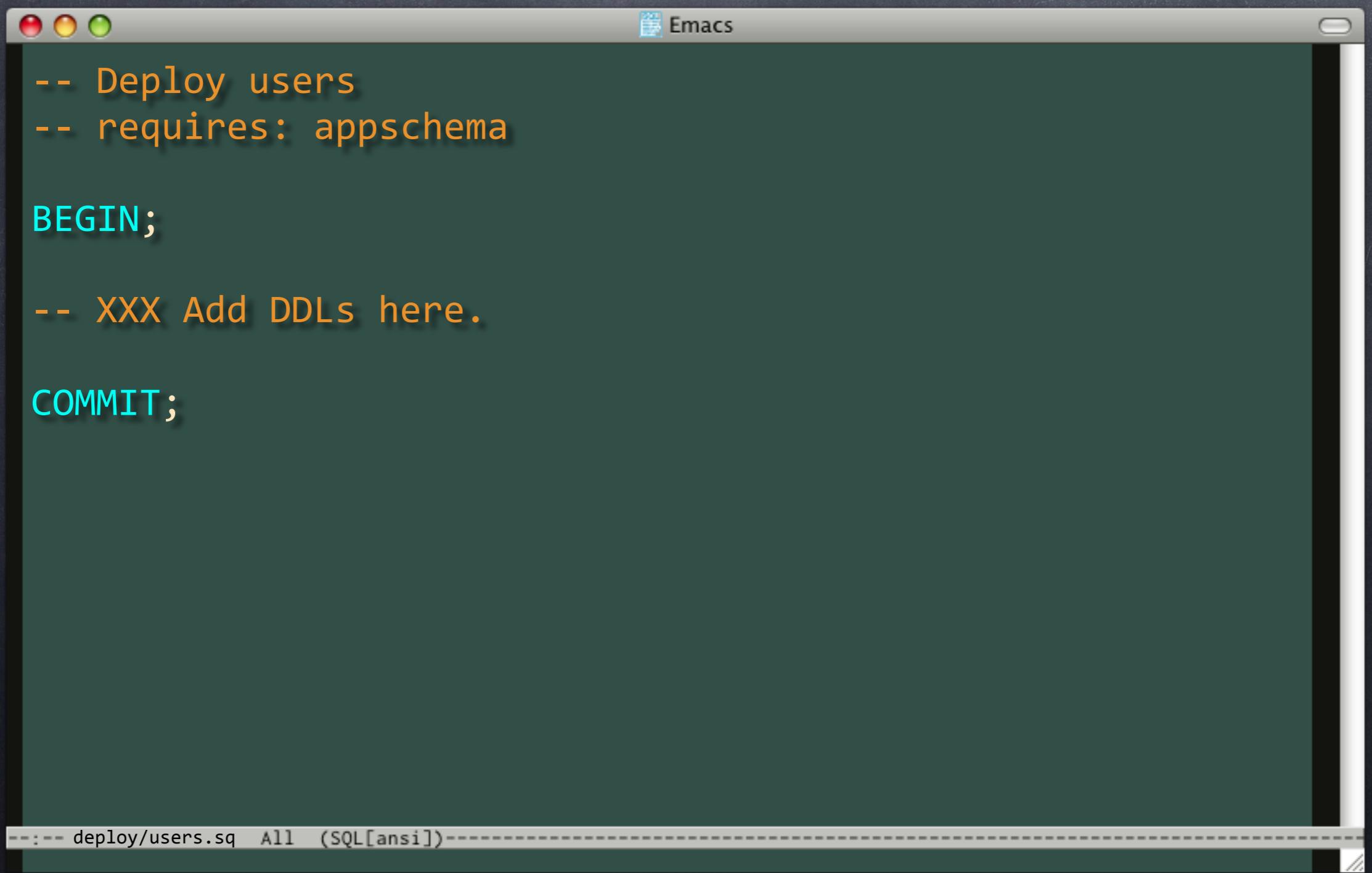
Dependencies!

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

deploy/users.sql



The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". The buffer contains the following SQL code:

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

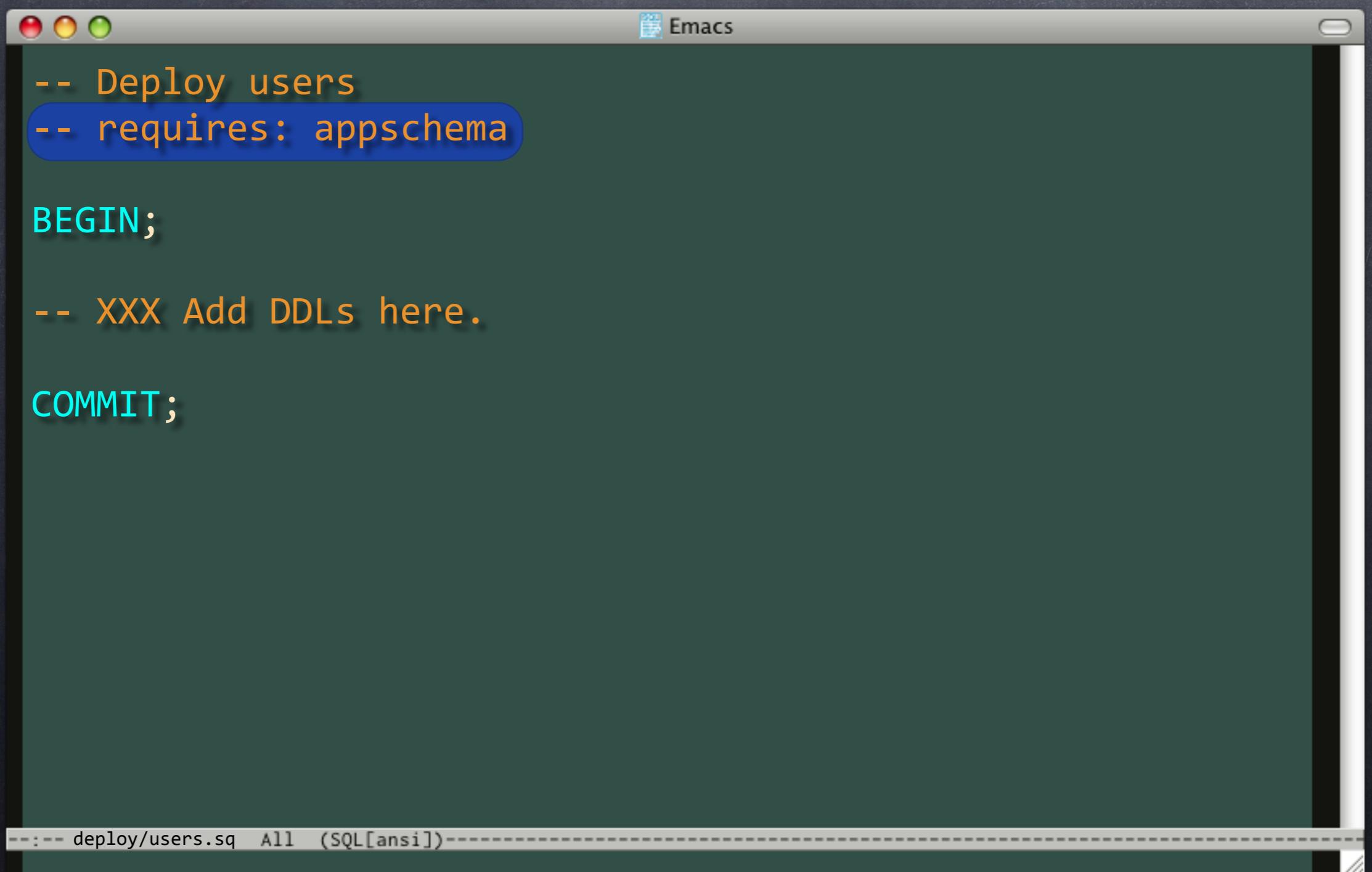
BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

At the bottom of the window, the status bar displays the path "---- deploy/users.sql All (SQL[ansi])----".

deploy/users.sql



The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". The buffer contains the following SQL code:

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

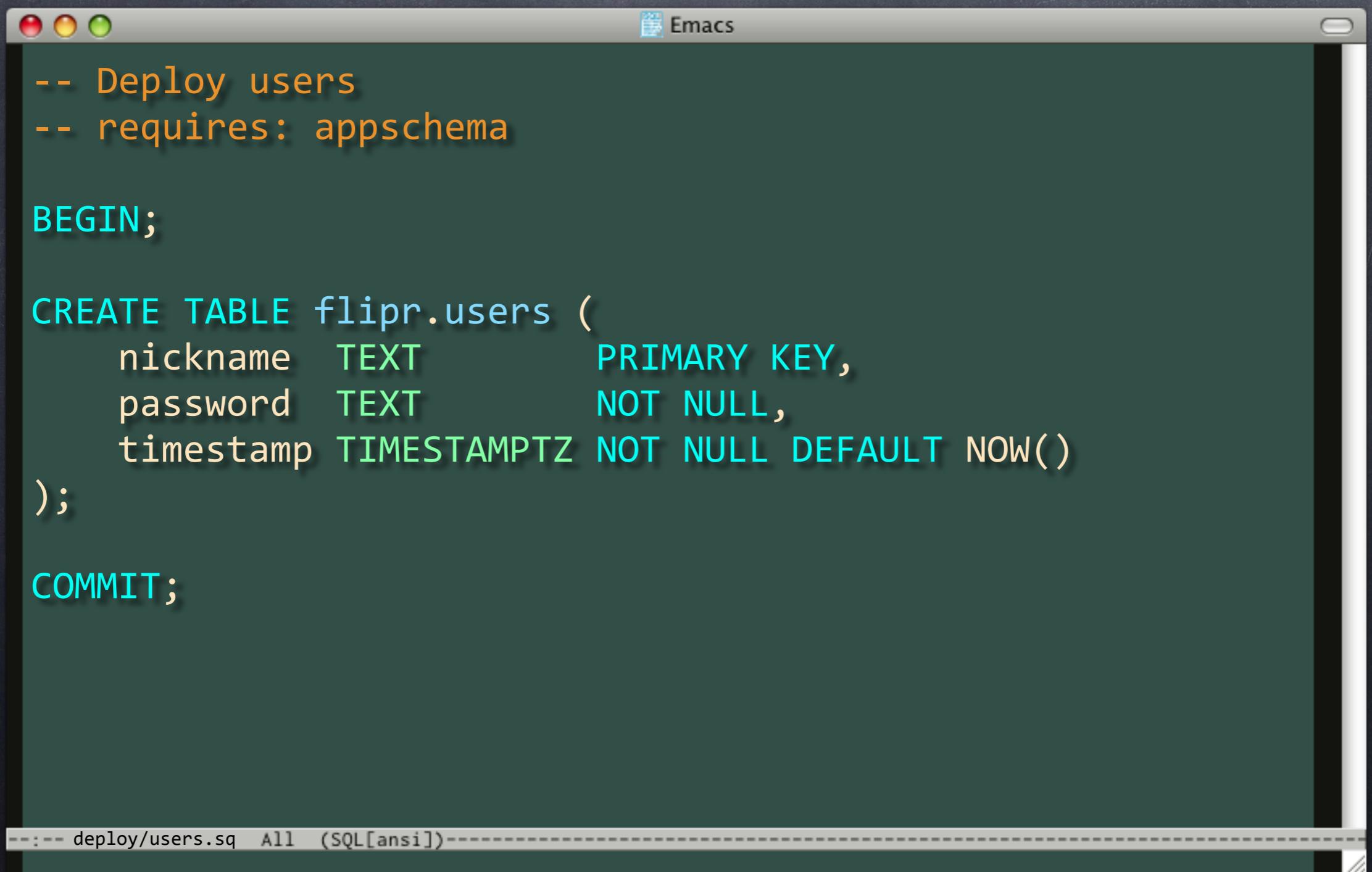
BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

At the bottom of the window, the status bar displays the path "---- deploy/users.sql All (SQL[ansi])----".

deploy/users.sql



The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". The buffer contains the following SQL code:

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

BEGIN;

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

COMMIT;
```

At the bottom of the window, the status bar displays the path "deploy/users.sql All (SQL[ansi])".

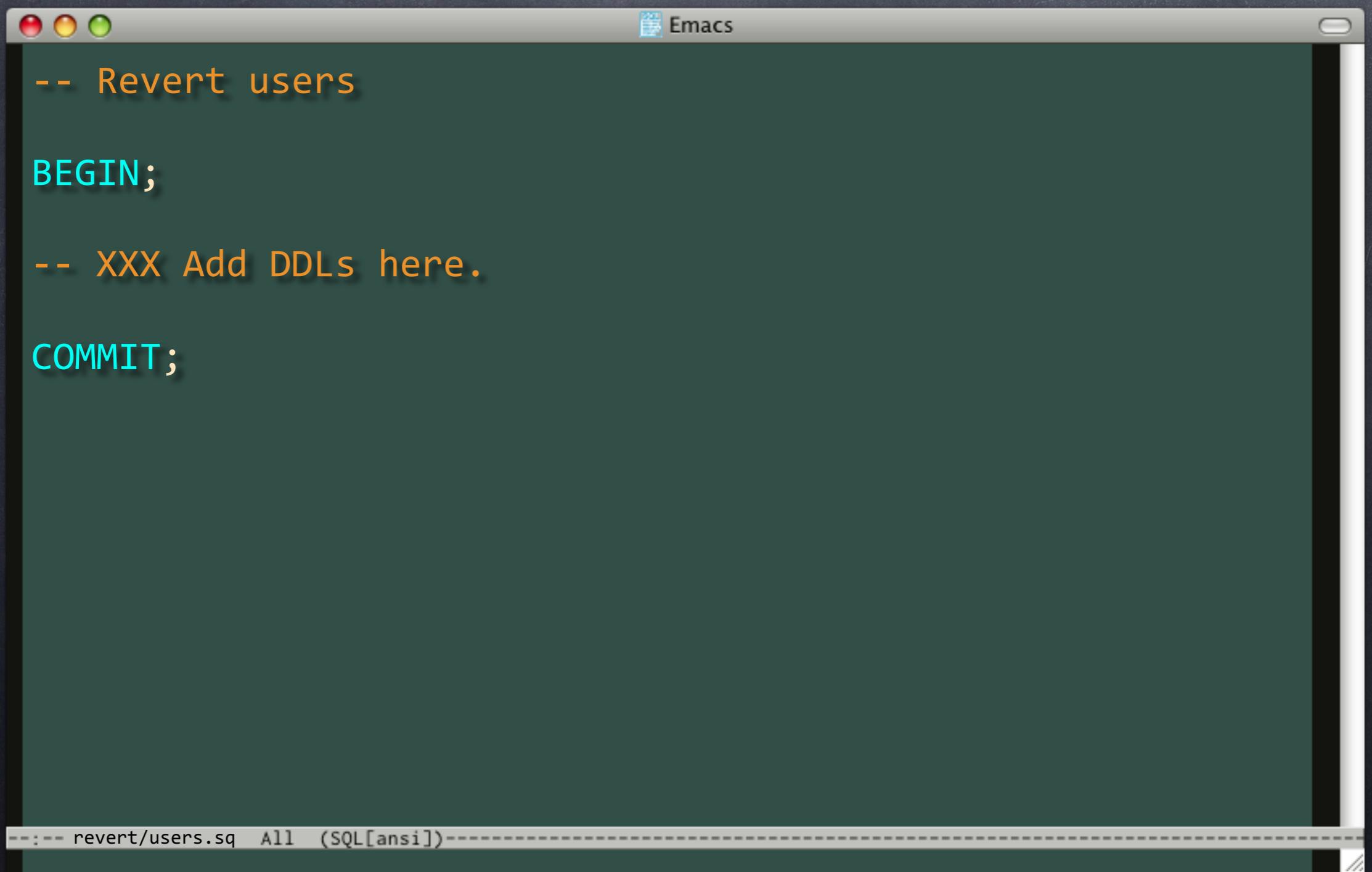
Dependencies!

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

revert/users.sql



The image shows a screenshot of an Emacs window with a dark green background. The window title is "Emacs". The buffer contains the following SQL code:

```
-- Revert users

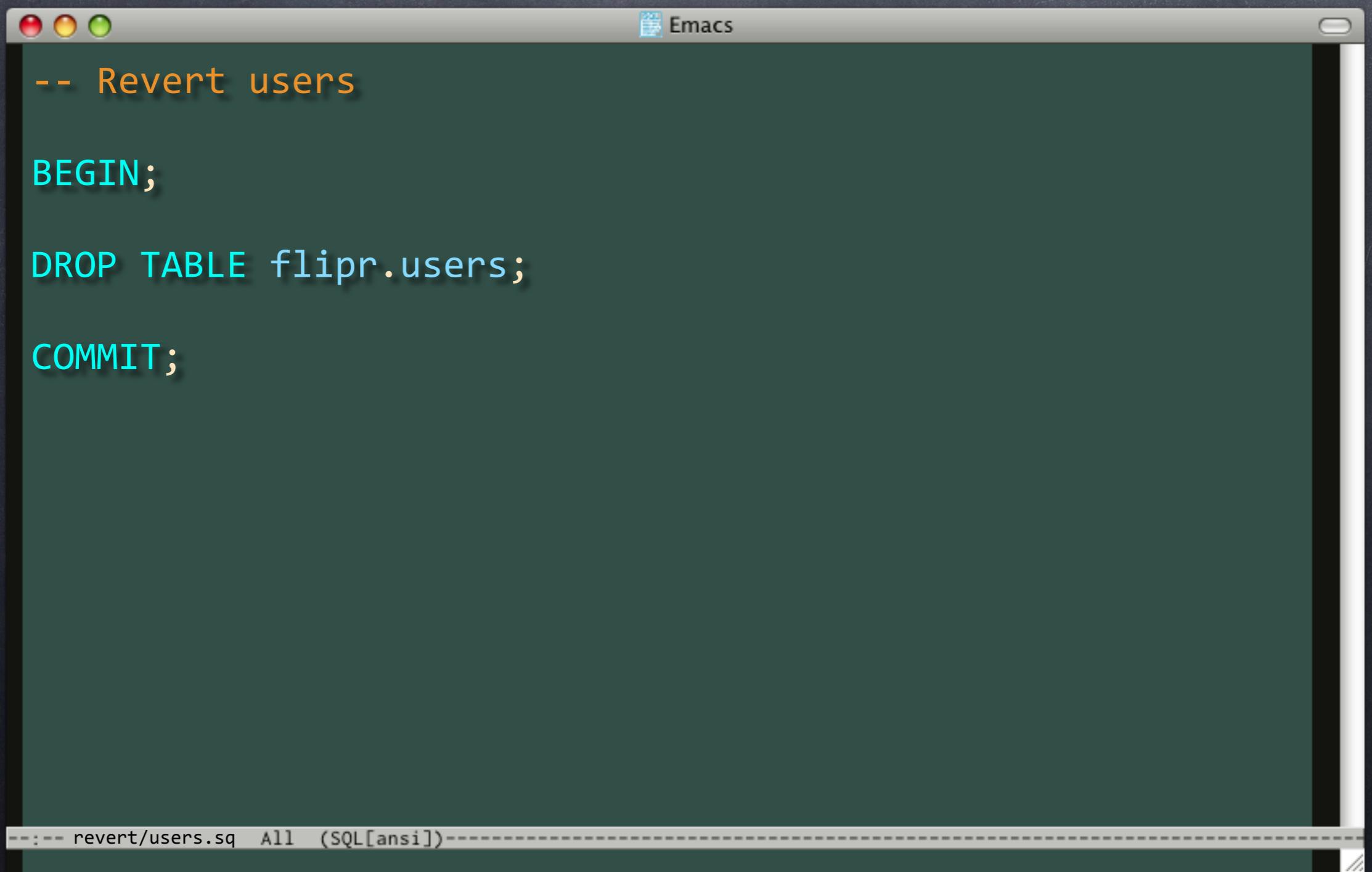
BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

At the bottom of the window, the status bar displays the path "revert/users.sql" and the mode "(SQL[ansi])".

revert/users.sql



The image shows a screenshot of an Emacs window with a dark green background. The window title is "Emacs". The buffer contains the following SQL code:

```
-- Revert users

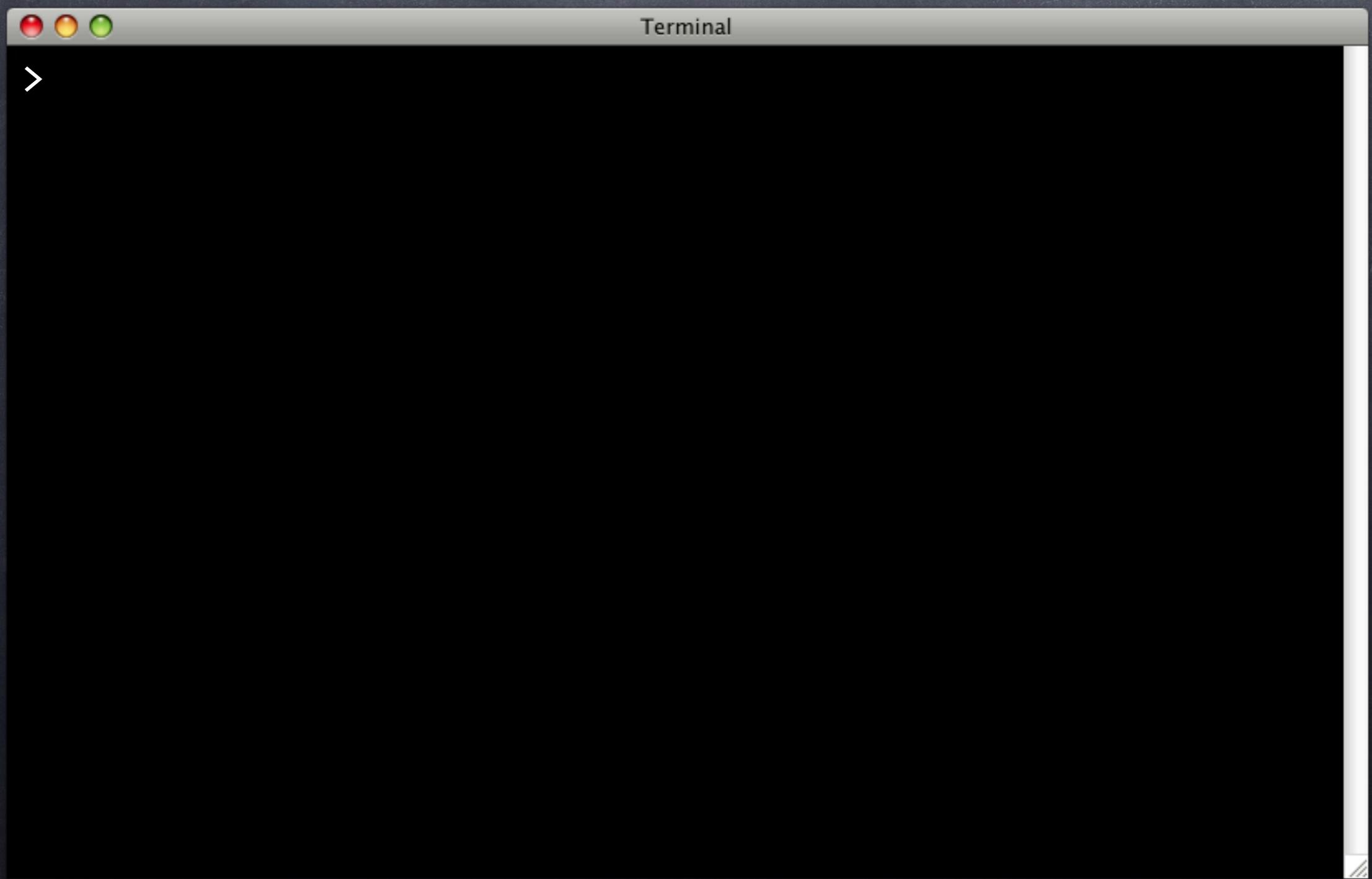
BEGIN;

DROP TABLE flipr.users;

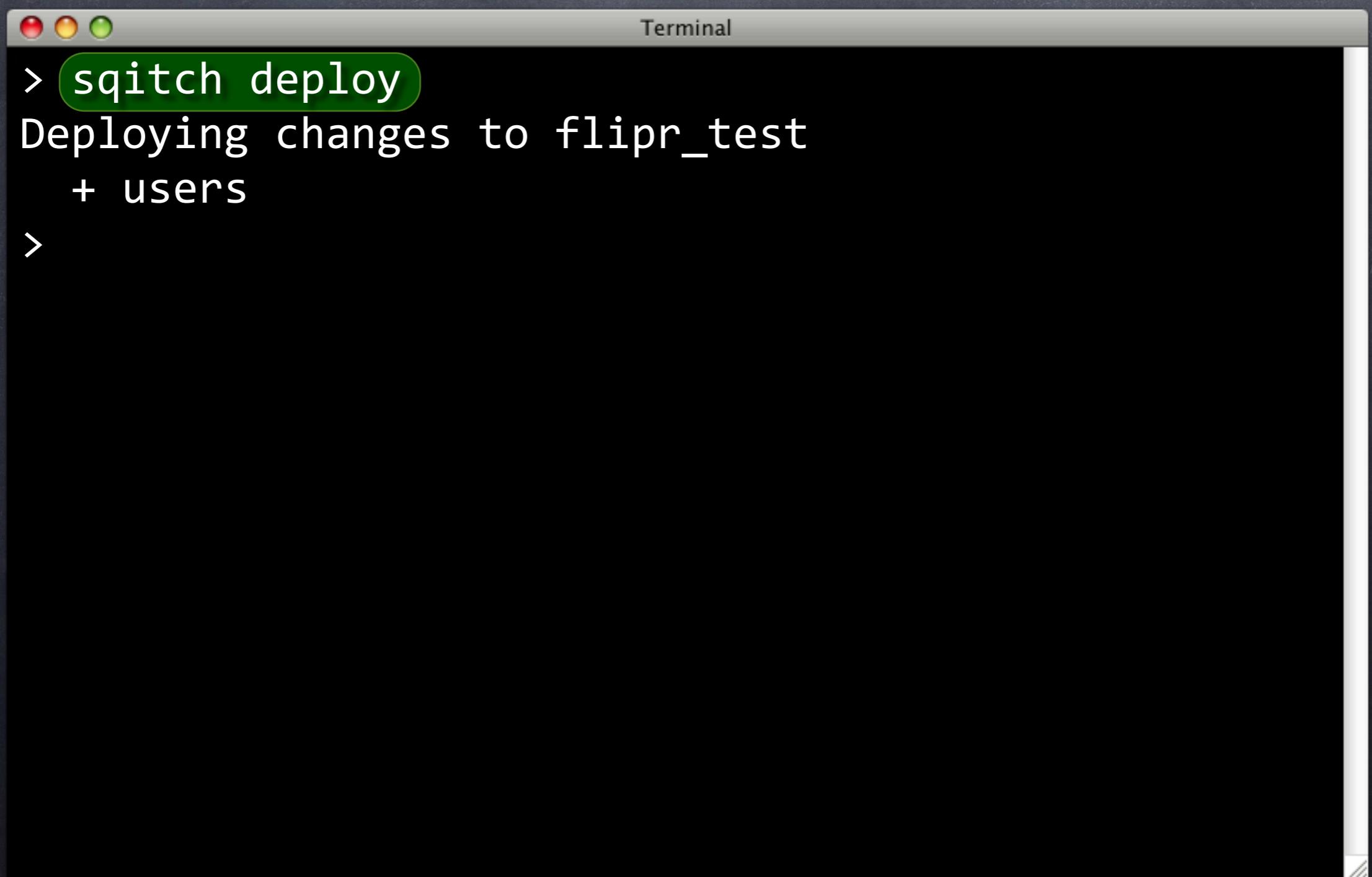
COMMIT;
```

At the bottom of the window, the status bar displays the path "revert/users.sql" and the mode "All (SQL[ansi])".

Make Users

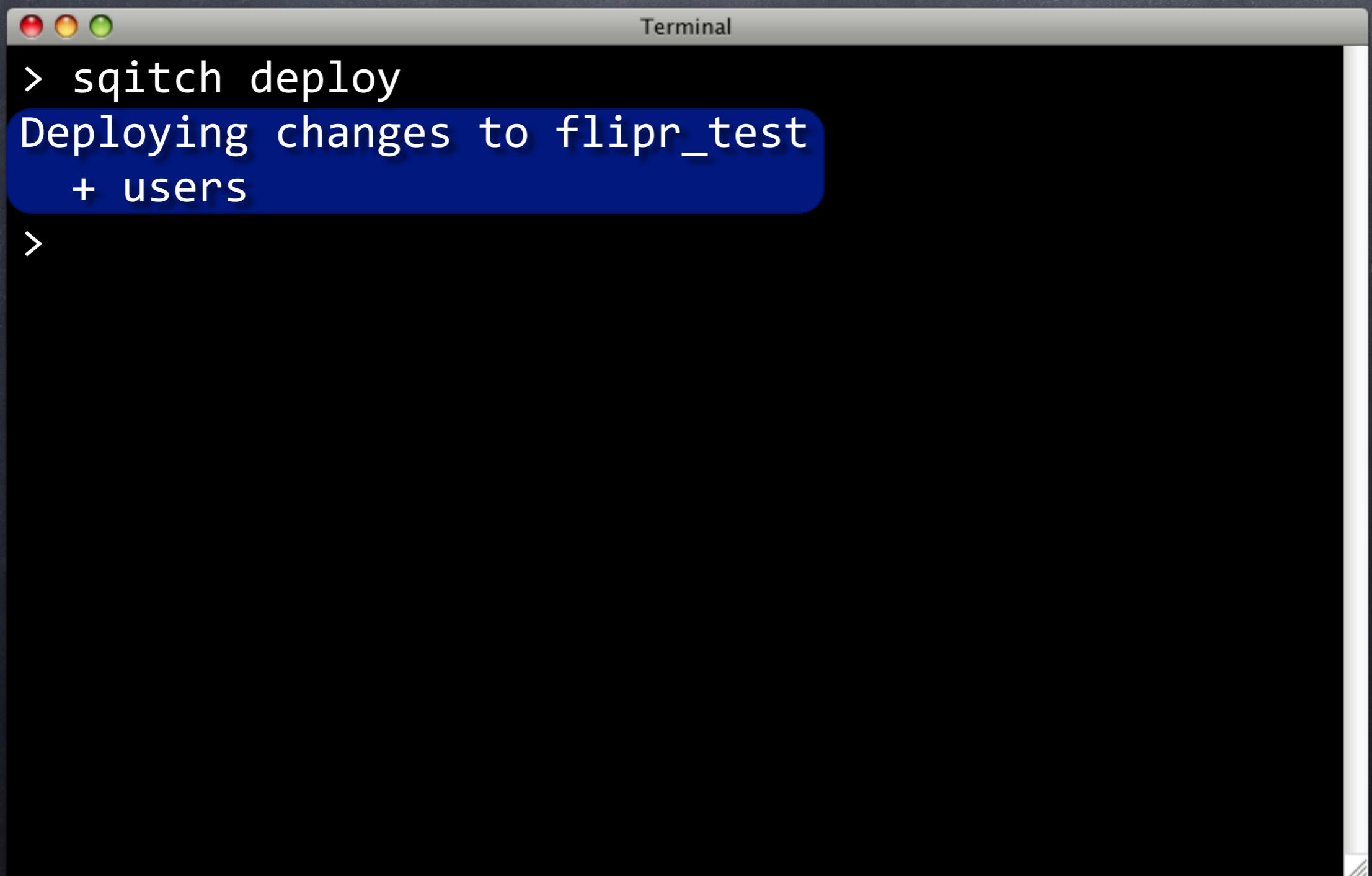


Make Users



```
Terminal  
> sqitch deploy  
Deploying changes to flipr_test  
+ users  
>
```

Make Users

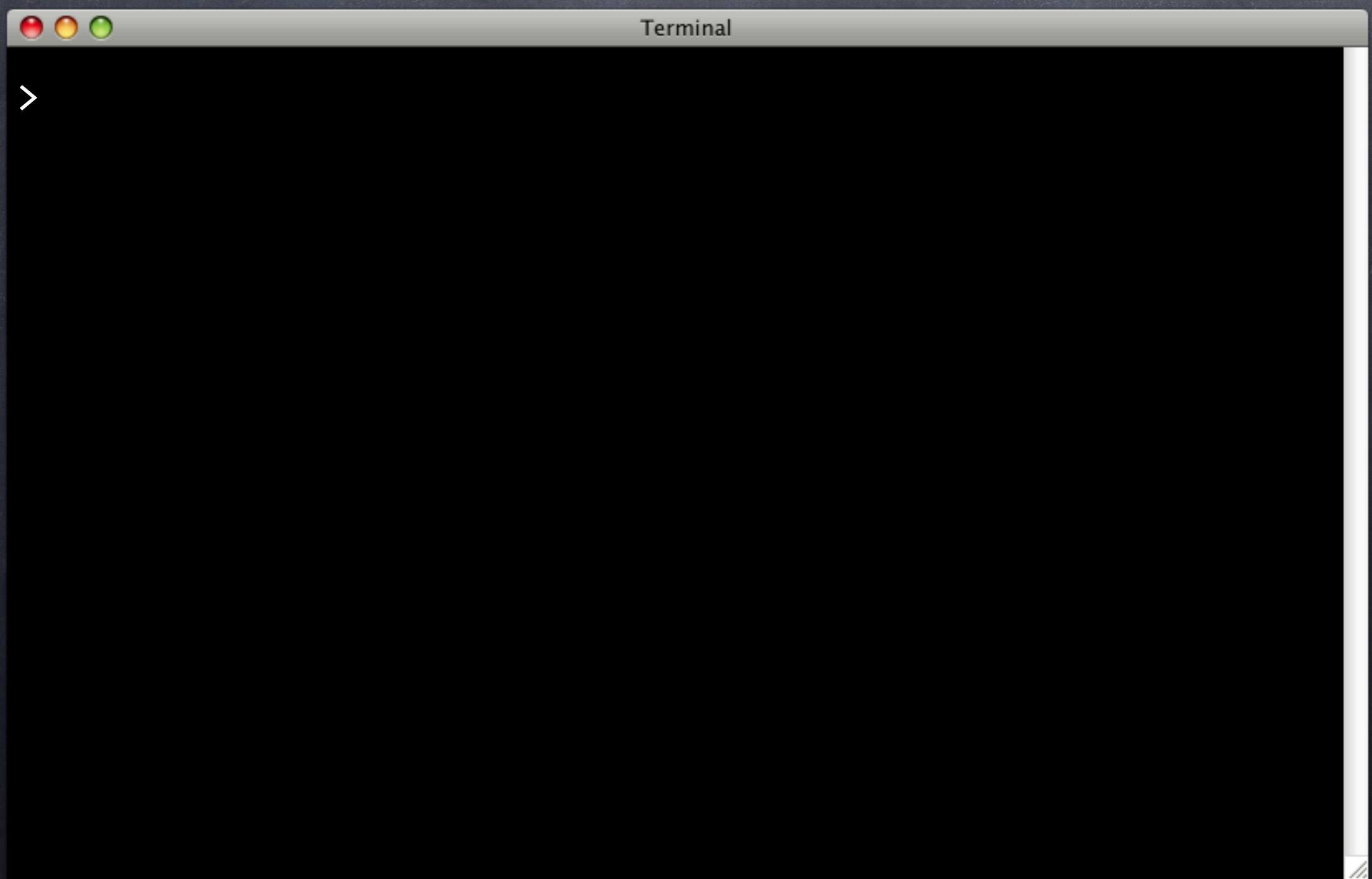


```
Terminal  
> sqitch deploy  
Deploying changes to flipr_test  
+ users  
>
```

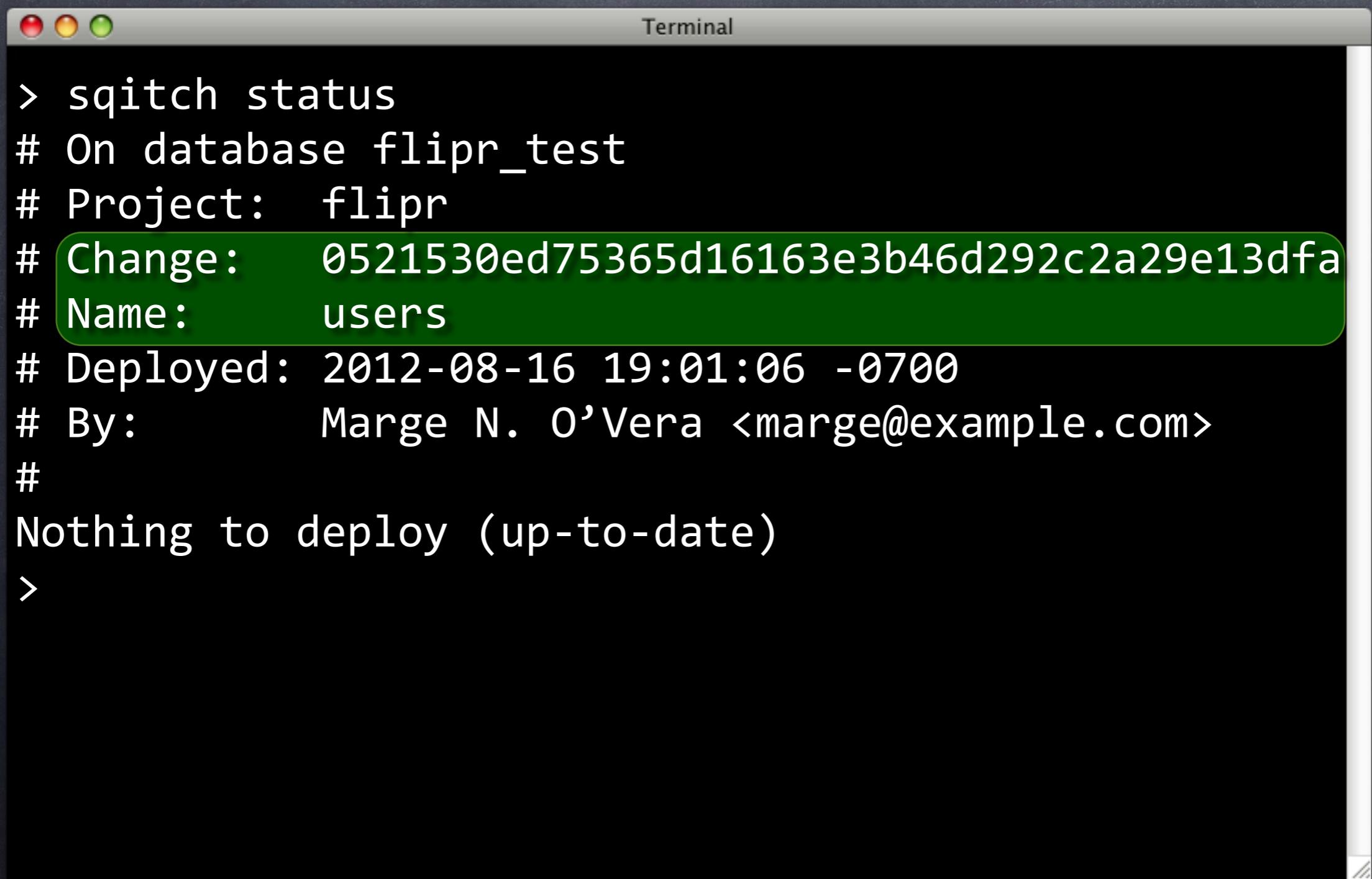
Make Users

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

Status Update



Status Update



A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the output of the command "sqitch status". The output shows details about a database named "flipr_test" under a project "flipr". A specific change set is highlighted with a green rounded rectangle, containing the "Change", "Name", and "Deployed" fields. The "By" field is also visible. The message "Nothing to deploy (up-to-date)" is at the bottom, and the prompt ">" is shown at the very end.

```
> sqitch status
# On database flipr_test
# Project: flipr
# Change: 0521530ed75365d16163e3b46d292c2a29e13dfa
# Name: users
# Deployed: 2012-08-16 19:01:06 -0700
# 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: 0521530ed75365d16163e3b46d292c2a29e13dfa
# Name: users
# Deployed: 2012-08-16 19:01:06 -0700
# 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: 0521530ed75365d16163e3b46d292c2a29e13dfa
# Name: users
# Deployed: 2012-08-16 19:01:06 -0700
# By: Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
> sqitch revert --to @LAST^
Reverting changes to appschema from flipr_test
- users
>
```

Status Update

```
Terminal

> sqitch status
# On database flipr_test
# Project: flipr
# Change: 0521530ed75365d16163e3b46d292c2a29e13dfa
# Name: users
# Deployed: 2012-08-16 19:01:06 -0700
# By: Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
> sqitch revert --to @LAST^
Reverting changes to appschema from flipr_test
- users
>
```

Symbolic Tags

Symbolic Tags

• @LAST

Last deployed change

Symbolic Tags

- @LAST Last deployed change
- @FIRST First deployed change

Symbolic Tags

- @LAST Last deployed change
- @FIRST First deployed change
- @HEAD Last change in the plan

Symbolic Tags

- ⦿ @LAST Last deployed change
- ⦿ @FIRST First deployed change
- ⦿ @HEAD Last change in the plan
- ⦿ @ROOT First change in the plan

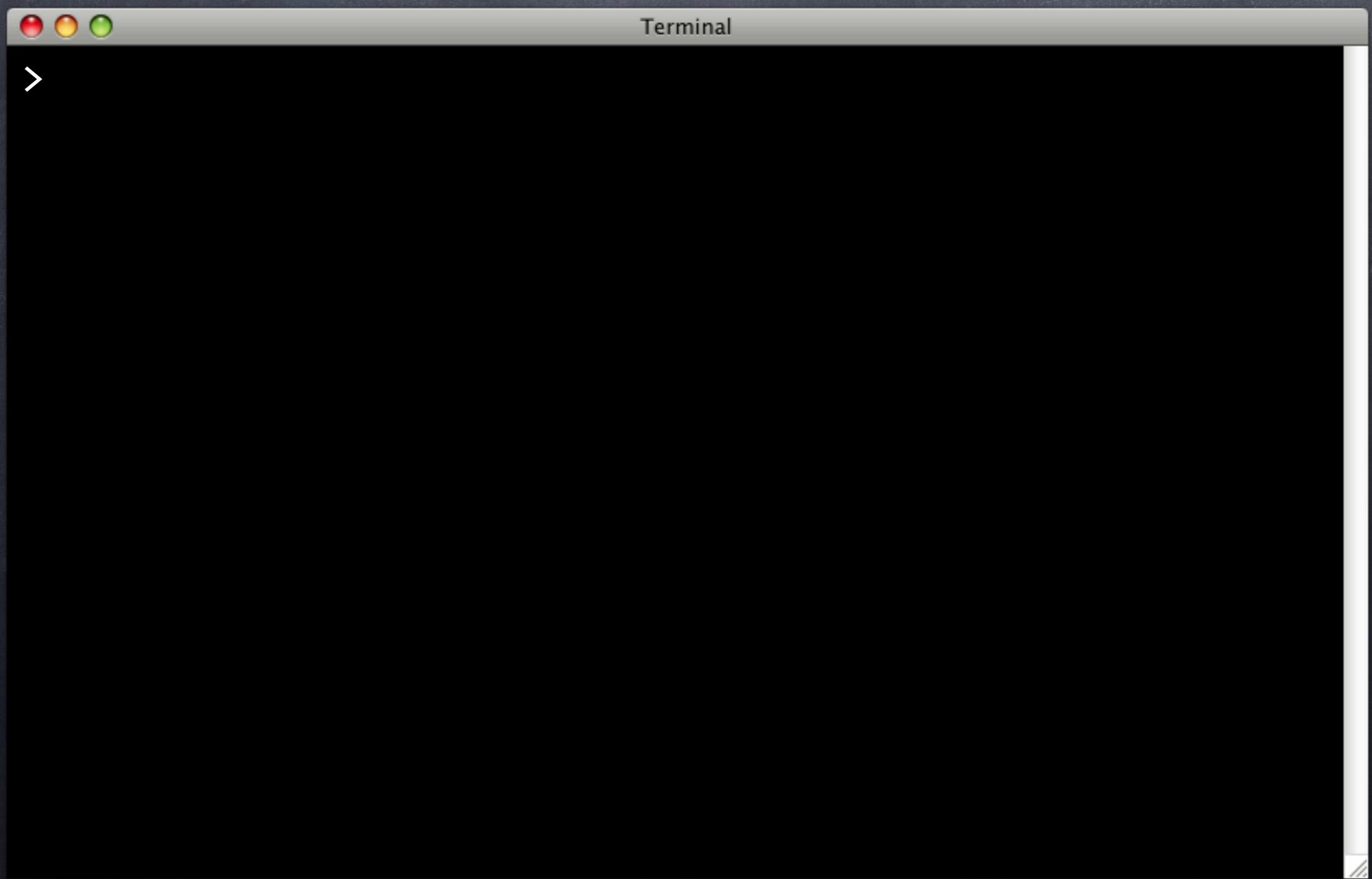
Symbolic Tags

- ⦿ @LAST Last deployed change
- ⦿ @FIRST First deployed change
- ⦿ @HEAD Last change in the plan
- ⦿ @ROOT First change in the plan
- ⦿ ^ Previous change

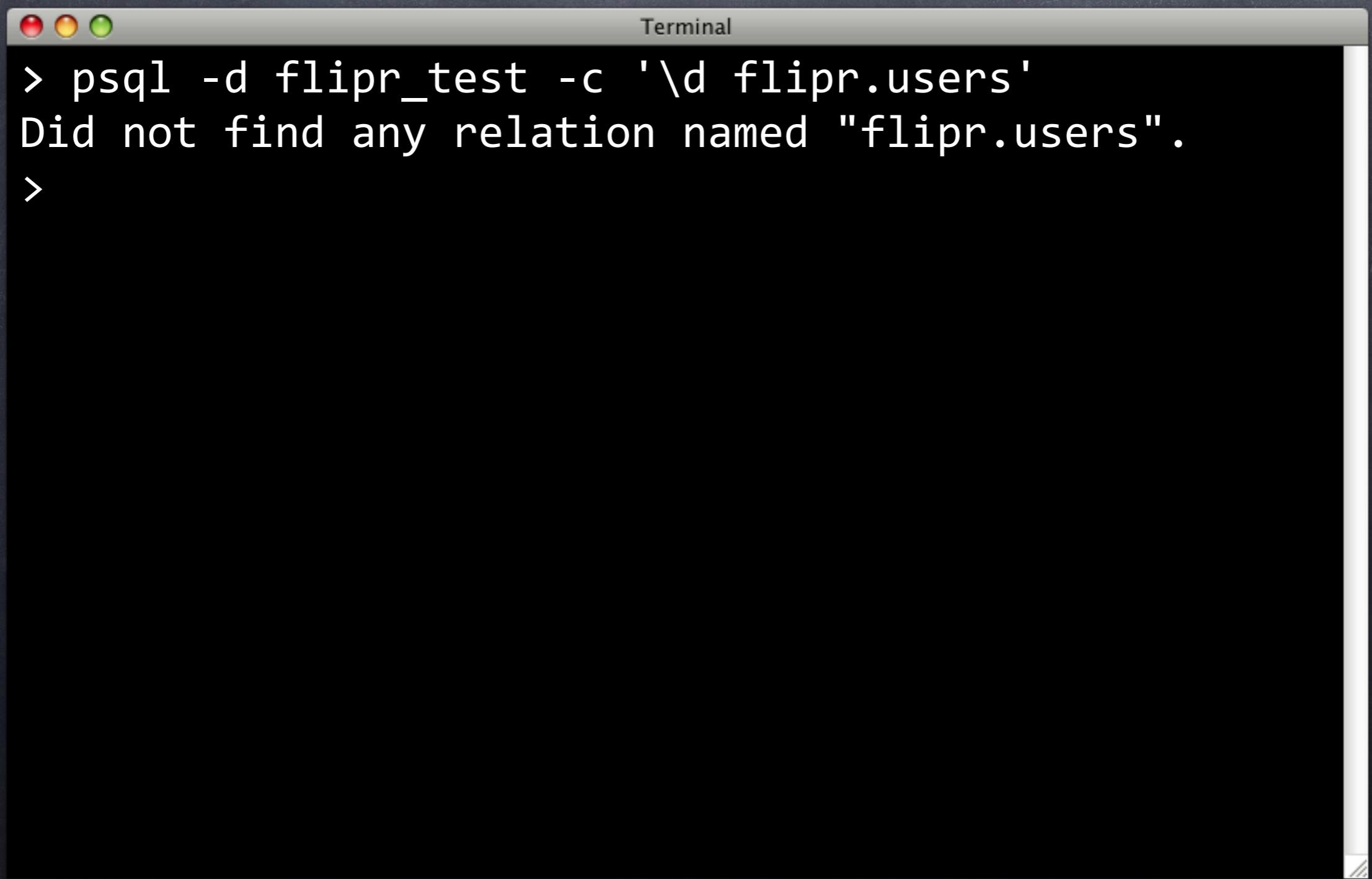
Symbolic Tags

⦿ @LAST	Last deployed change
⦿ @FIRST	First deployed change
⦿ @HEAD	Last change in the plan
⦿ @ROOT	First change in the plan
⦿ ^	Previous change
⦿ ~	Following change

Lose Users



Lose Users



A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The main area contains the following text:

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

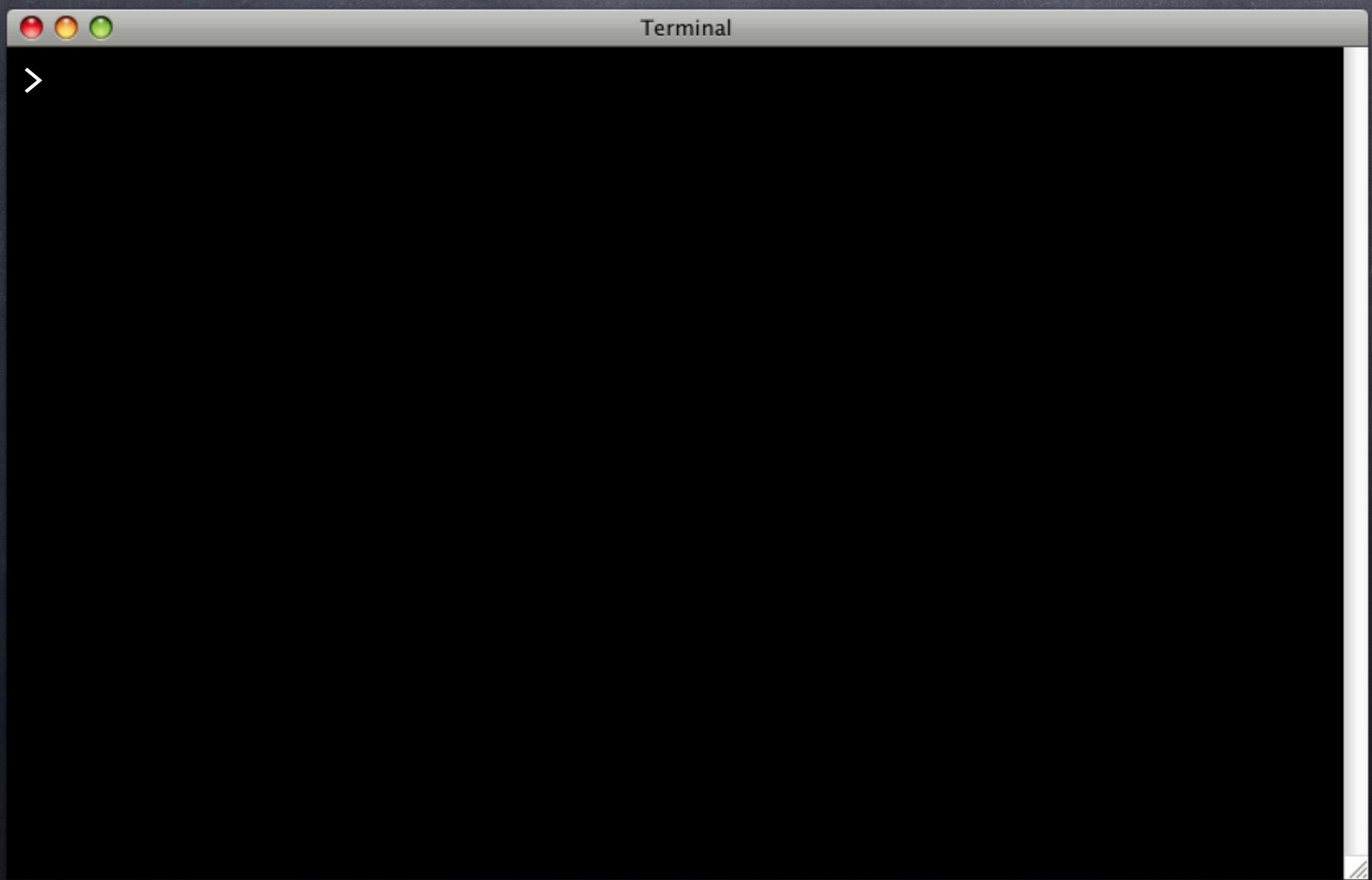
Lose Users

```
Terminal  
> psql -d flipr_test -c '\d flipr.users'  
Did not find any relation named "flipr.users".  
> git add .  
> git commit -am 'Add users table.'  
[master dfa6952] Add users table.  
 5 files changed, 30 insertions(+)  
 create mode 100644 deploy/users.sql  
 create mode 100644 revert/users.sql  
 create mode 100644 test/users.sql  
>
```

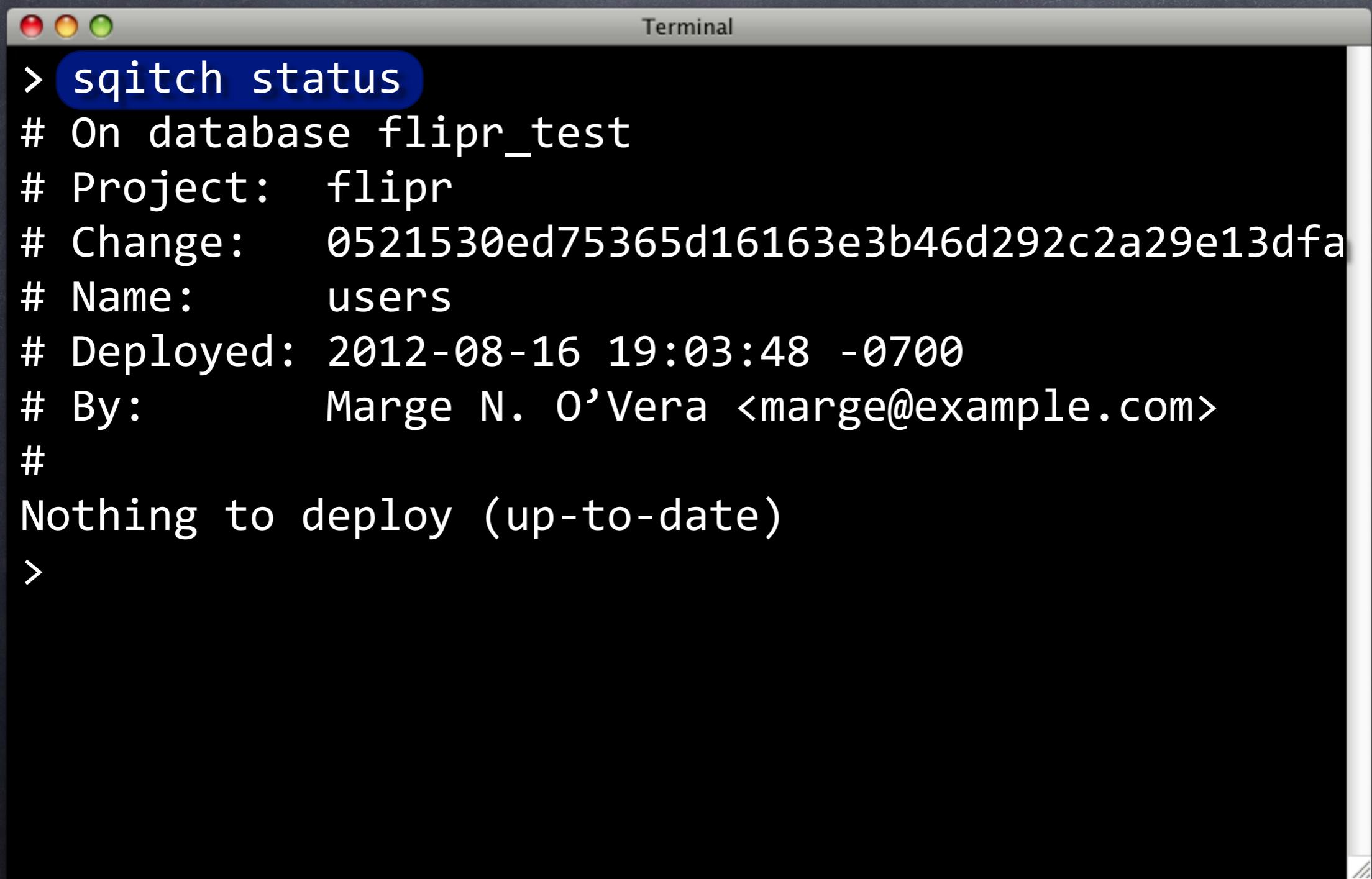
Lose Users

```
Terminal  
> psql -d flipr_test -c '\d flipr.users'  
Did not find any relation named "flipr.users".  
> git add .  
> git commit -am 'Add users table.'  
[master dfa6952] Add users table.  
 5 files changed, 30 insertions(+)  
 create mode 100644 deploy/users.sql  
 create mode 100644 revert/users.sql  
 create mode 100644 test/users.sql  
> sqitch deploy  
Deploying changes to flipr_test  
 + users  
>
```

Up to Date



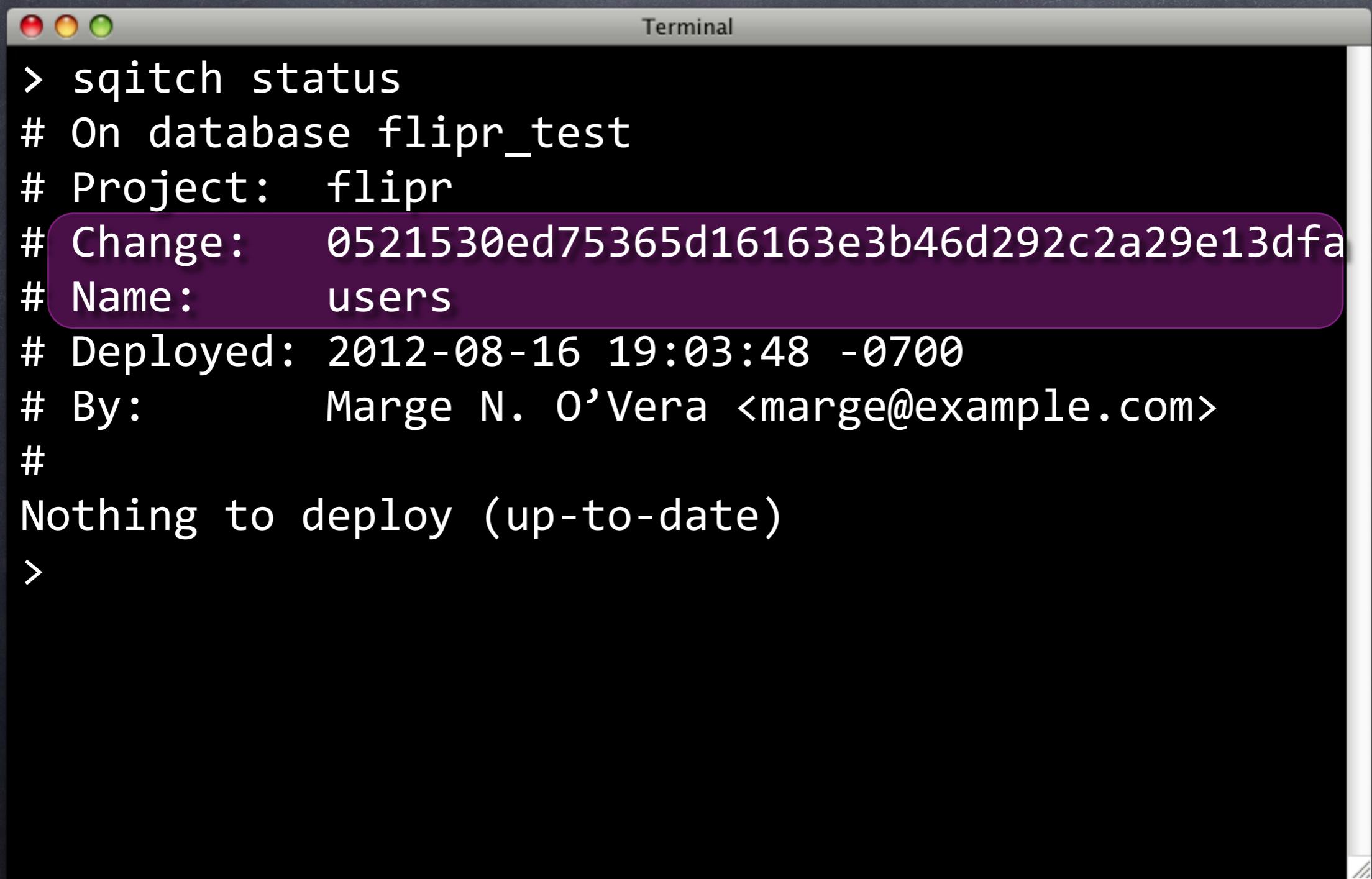
Up to Date



A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the output of the command "sqitch status". The output shows details about a database named "flipr_test" under project "flipr", with a specific change ID "0521530ed75365d16163e3b46d292c2a29e13dfa". It lists a single deployment entry for the "users" table by "Marge N. O'Vera <marge@example.com>" on "2012-08-16 19:03:48 -0700". The message "Nothing to deploy (up-to-date)" is displayed at the bottom, and the prompt ">" is shown at the very end.

```
> sqitch status
# On database flipr_test
# Project: flipr
# Change: 0521530ed75365d16163e3b46d292c2a29e13dfa
# Name: users
# Deployed: 2012-08-16 19:03:48 -0700
# By: Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

Up to Date



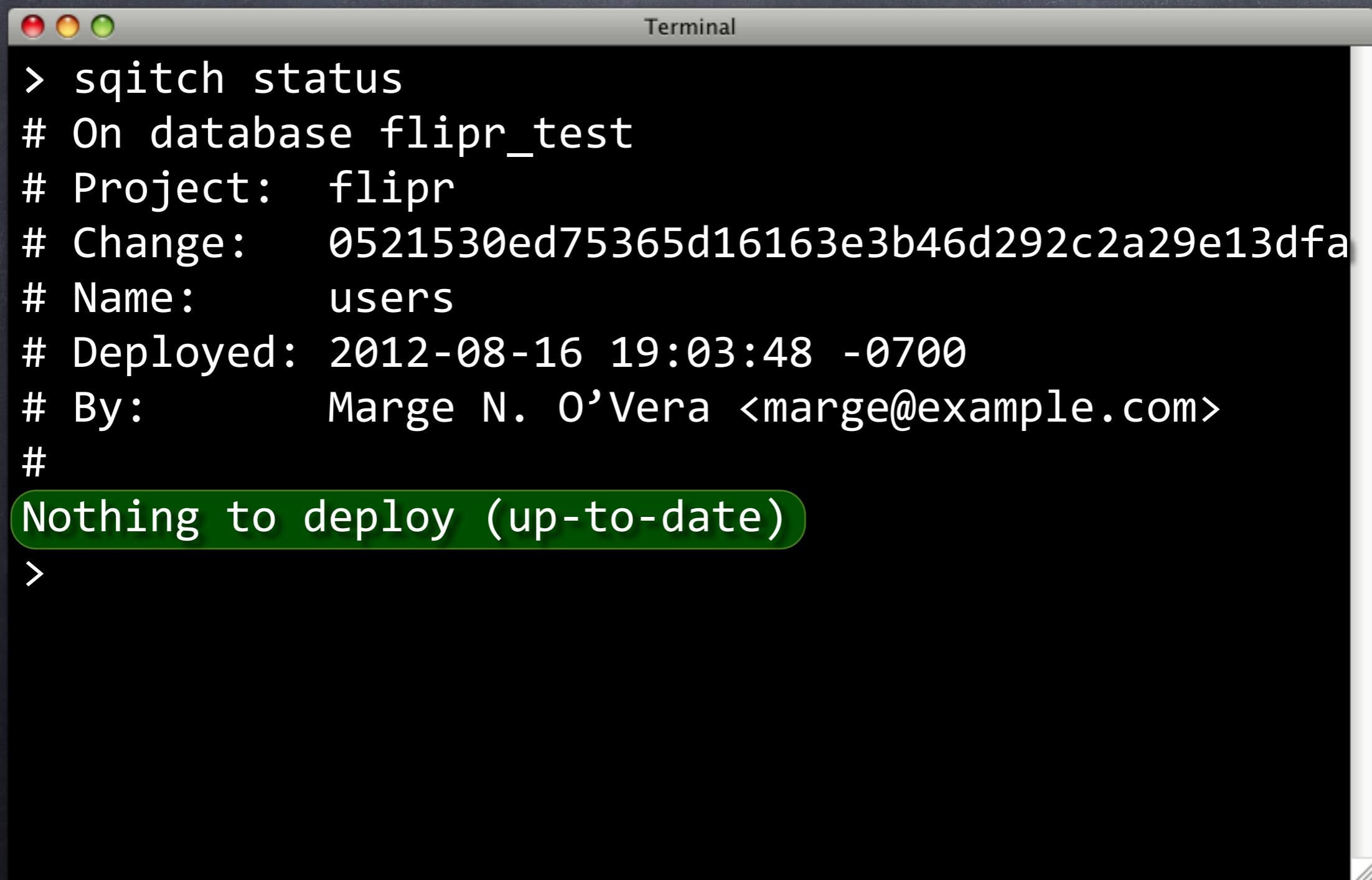
A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the output of the command "sqitch status". The output shows the following details for a database named "flipr_test" under project "flipr":

- # On database flipr_test
- # Project: flipr
- # Change: 0521530ed75365d16163e3b46d292c2a29e13dfa
- # Name: users
- # Deployed: 2012-08-16 19:03:48 -0700
- # By: Marge N. O'Vera <marge@example.com>
- #

The text "Nothing to deploy (up-to-date)" is displayed, followed by a final prompt ">". A purple rounded rectangle highlights the "Change:" line.

```
> sqitch status
# On database flipr_test
# Project: flipr
# Change: 0521530ed75365d16163e3b46d292c2a29e13dfa
# Name: users
# Deployed: 2012-08-16 19:03:48 -0700
# By: Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

Up to Date

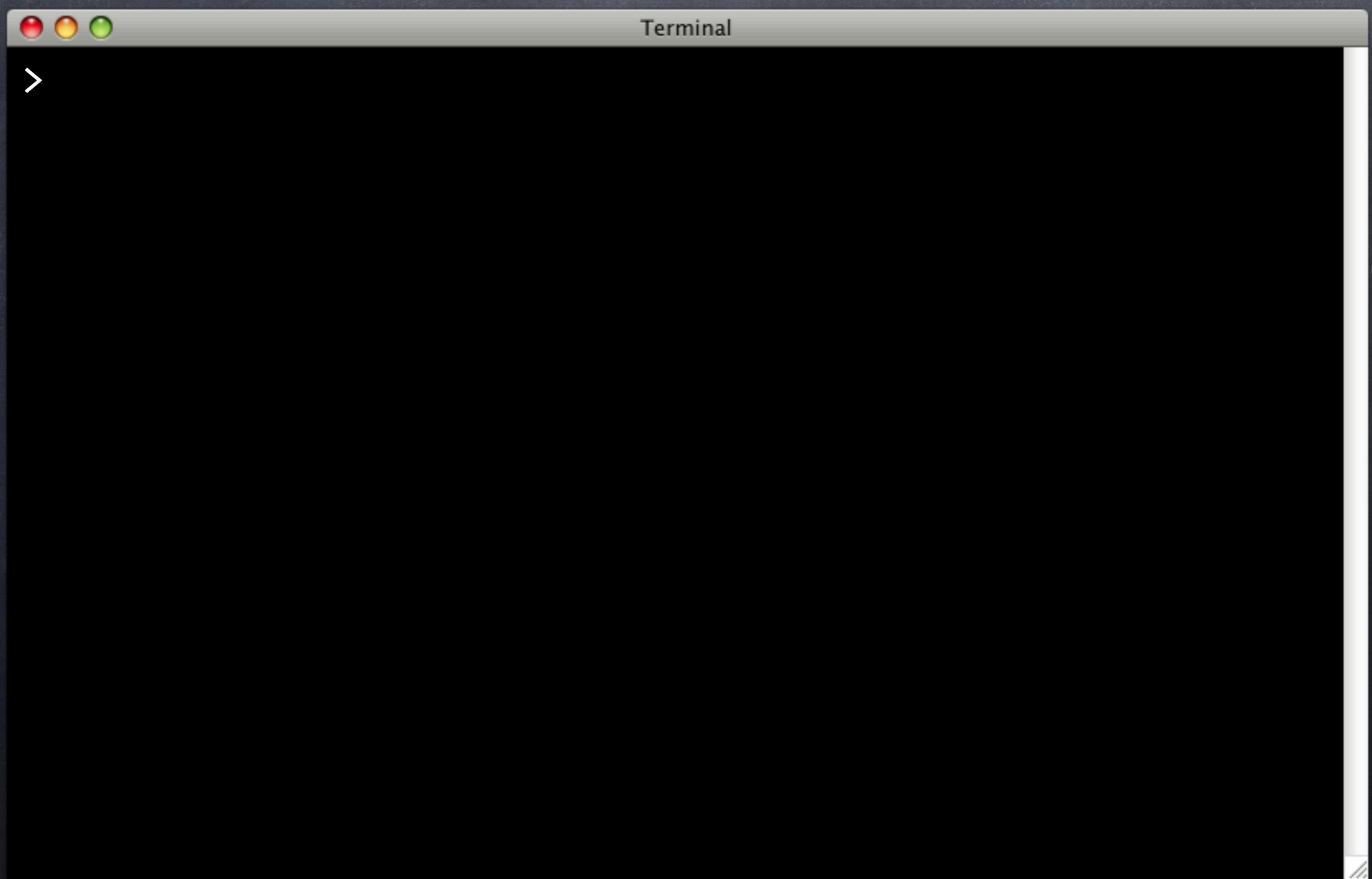


A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the following text output from the command "sqitch status":

```
> sqitch status
# On database flipr_test
# Project: flipr
# Change: 0521530ed75365d16163e3b46d292c2a29e13dfa
# Name: users
# Deployed: 2012-08-16 19:03:48 -0700
# By: Marge N. O'Vera <marge@example.com>
#
Nothing to deploy (up-to-date)
>
```

The text "Nothing to deploy (up-to-date)" is highlighted with a green rounded rectangle.

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 test/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 test/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 test/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 test/change_pass.sql  
Added "change_pass [users appschema]" to sqitch.plan  
  
>
```

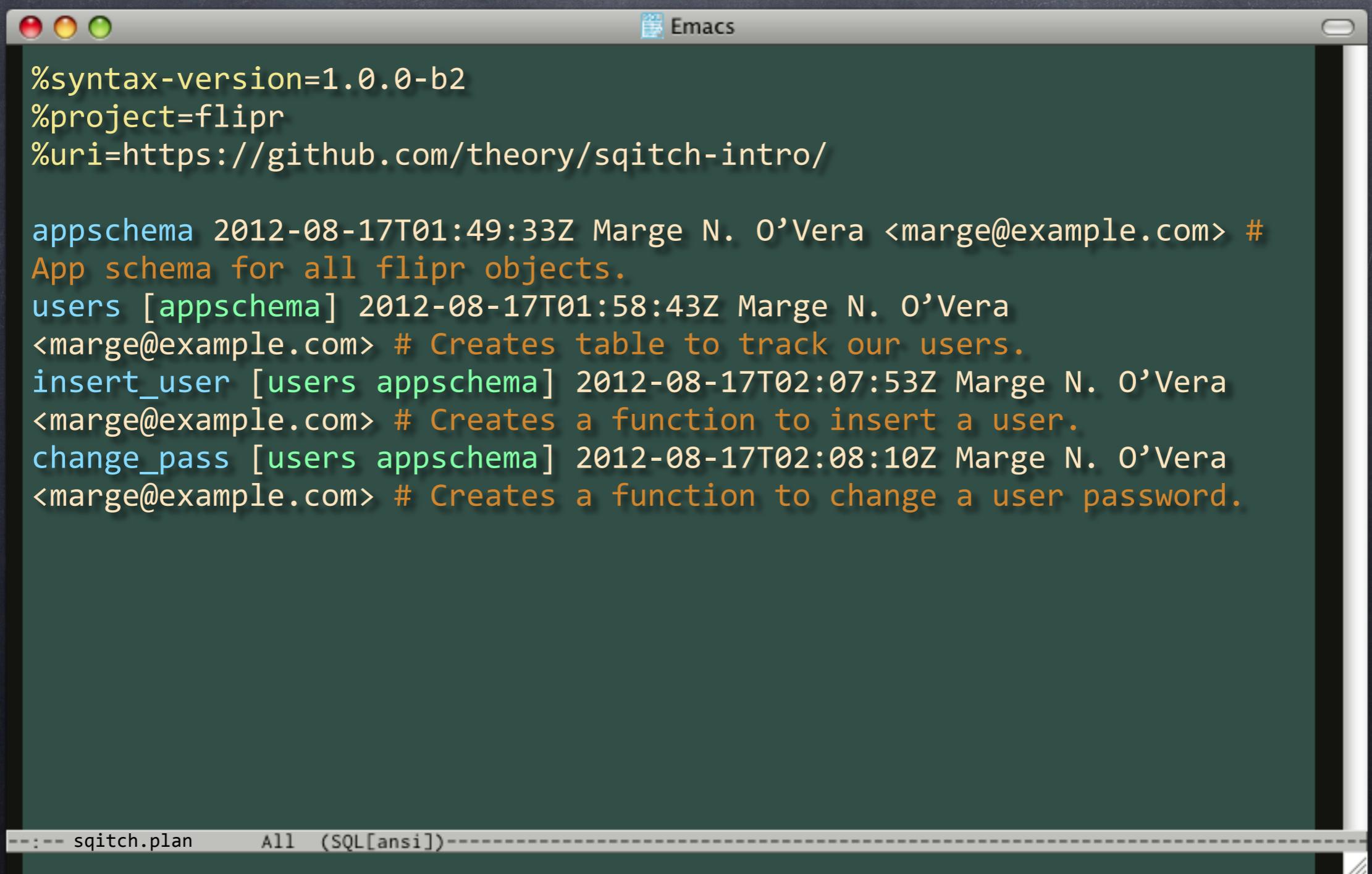
A Twofer

```
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 test/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 test/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 test/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 test/change_pass.sql  
Added "change_pass [users appschema]" to sqitch.plan  
  
> emacs sqitch.plan
```

sqitch.plan



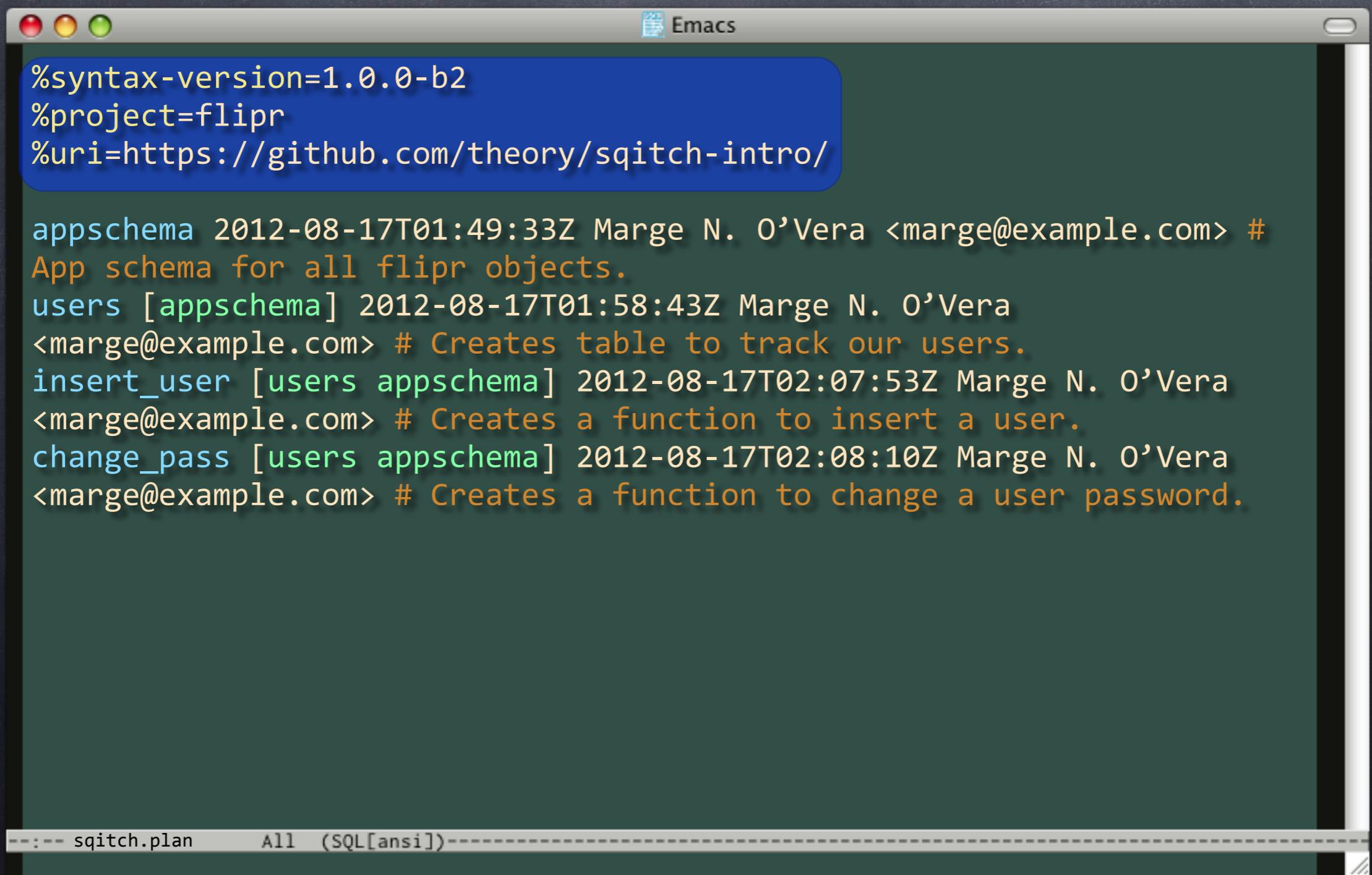
The screenshot shows an Emacs window with a dark green background and white text. The title bar reads "Emacs". The buffer contains a sqitch.plan file with the following content:

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

appschema 2012-08-17T01:49:33Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-08-17T01:58:43Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-08-17T02:07:53Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-08-17T02:08:10Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.

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

sqitch.plan



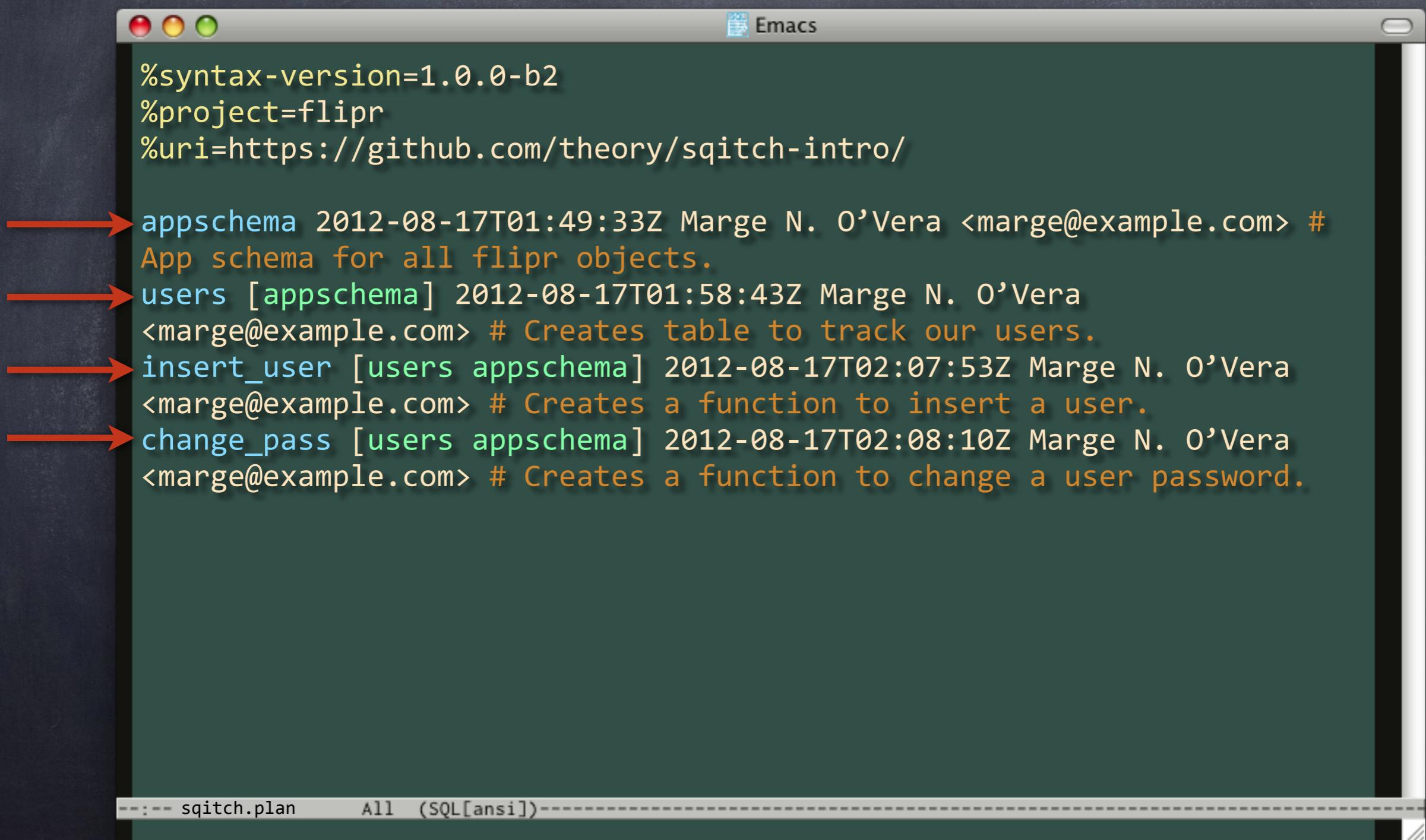
The screenshot shows an Emacs window with a dark theme, displaying a sqitch.plan file. The window title is "Emacs". The buffer contains the following SQL script:

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

appschema 2012-08-17T01:49:33Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-08-17T01:58:43Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-08-17T02:07:53Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-08-17T02:08:10Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.

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

sqitch.plan



The screenshot shows an Emacs window with a dark green background and white text. The title bar says "Emacs". The buffer contains a sqitch.plan file with the following content:

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

→ appschema 2012-08-17T01:49:33Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
→ users [appschema] 2012-08-17T01:58:43Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
→ insert_user [users appschema] 2012-08-17T02:07:53Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
→ change_pass [users appschema] 2012-08-17T02:08:10Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
```

At the bottom of the window, there is a status bar with the text "sqitch.plan All (SQL[ansi])".

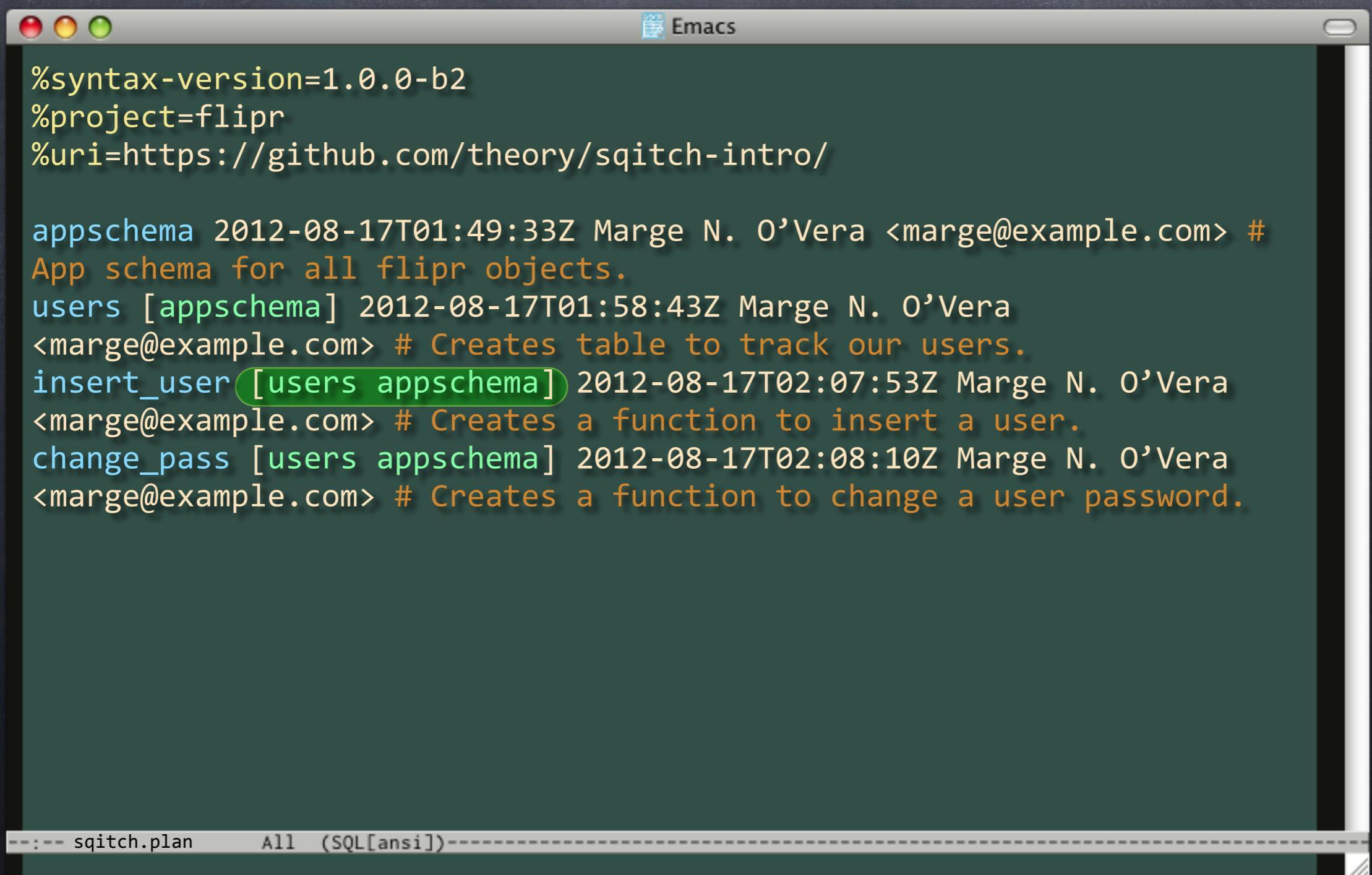
sqitch.plan

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

appschema 2012-08-17T01:49:33Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-08-17T01:58:43Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-08-17T02:07:53Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-08-17T02:08:10Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.

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

sqitch.plan



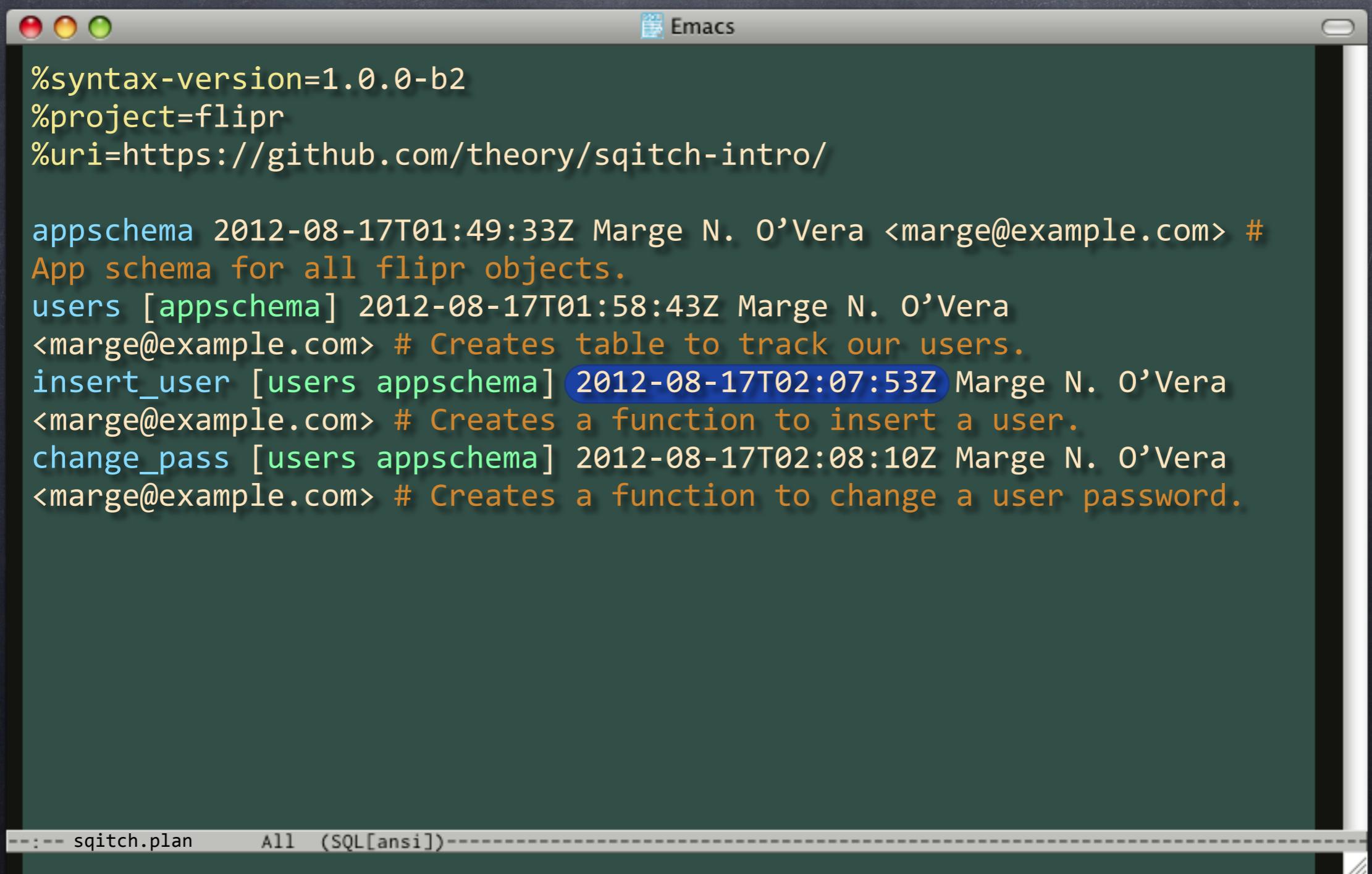
The screenshot shows an Emacs window with a dark green background and white text. The title bar reads "Emacs". The buffer contains a sqitch.plan file with the following content:

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

appschema 2012-08-17T01:49:33Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-08-17T01:58:43Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-08-17T02:07:53Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-08-17T02:08:10Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.

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

sqitch.plan



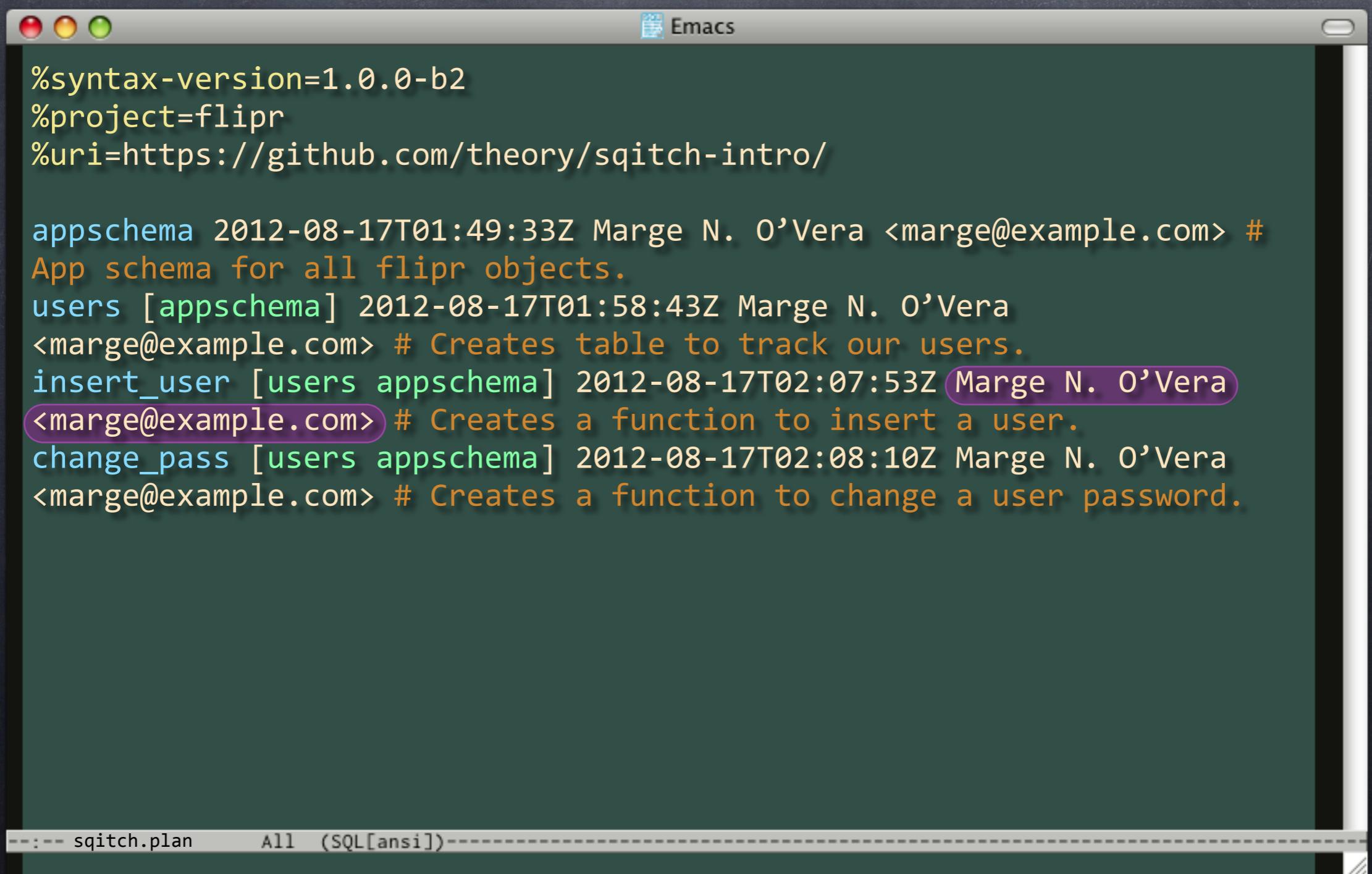
The screenshot shows an Emacs window with a dark green background and white text. The title bar reads "Emacs". The buffer contains a sqitch.plan file with the following content:

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

appschema 2012-08-17T01:49:33Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-08-17T01:58:43Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-08-17T02:07:53Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-08-17T02:08:10Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.

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

sqitch.plan



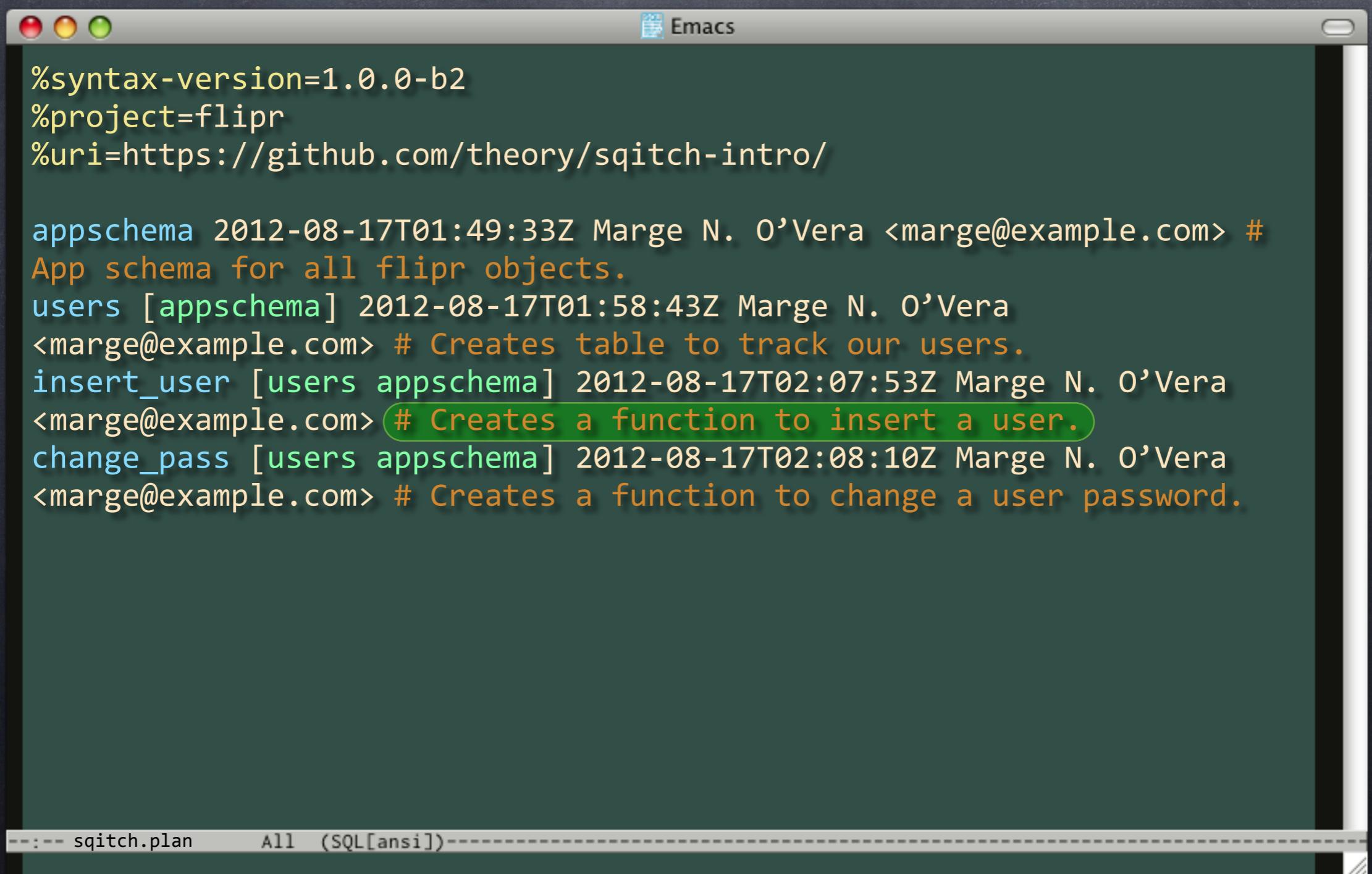
The screenshot shows an Emacs window with a dark theme, displaying a sqitch.plan file. The window title is "Emacs". The code in the buffer is as follows:

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

appschema 2012-08-17T01:49:33Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-08-17T01:58:43Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-08-17T02:07:53Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-08-17T02:08:10Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.

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

sqitch.plan



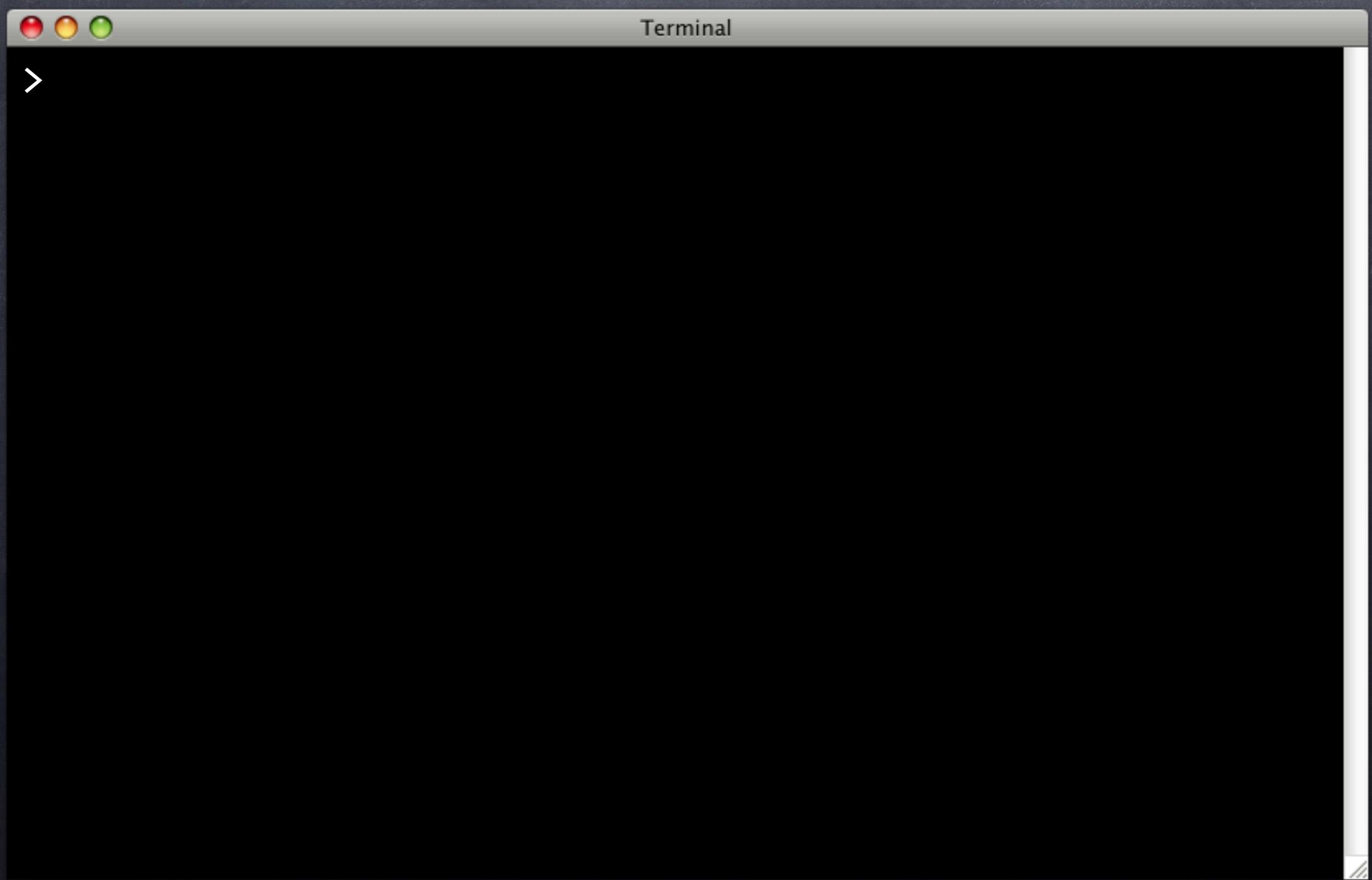
The screenshot shows an Emacs window with a dark green background and white text. The title bar reads "Emacs". The buffer contains a sqitch.plan file with the following content:

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

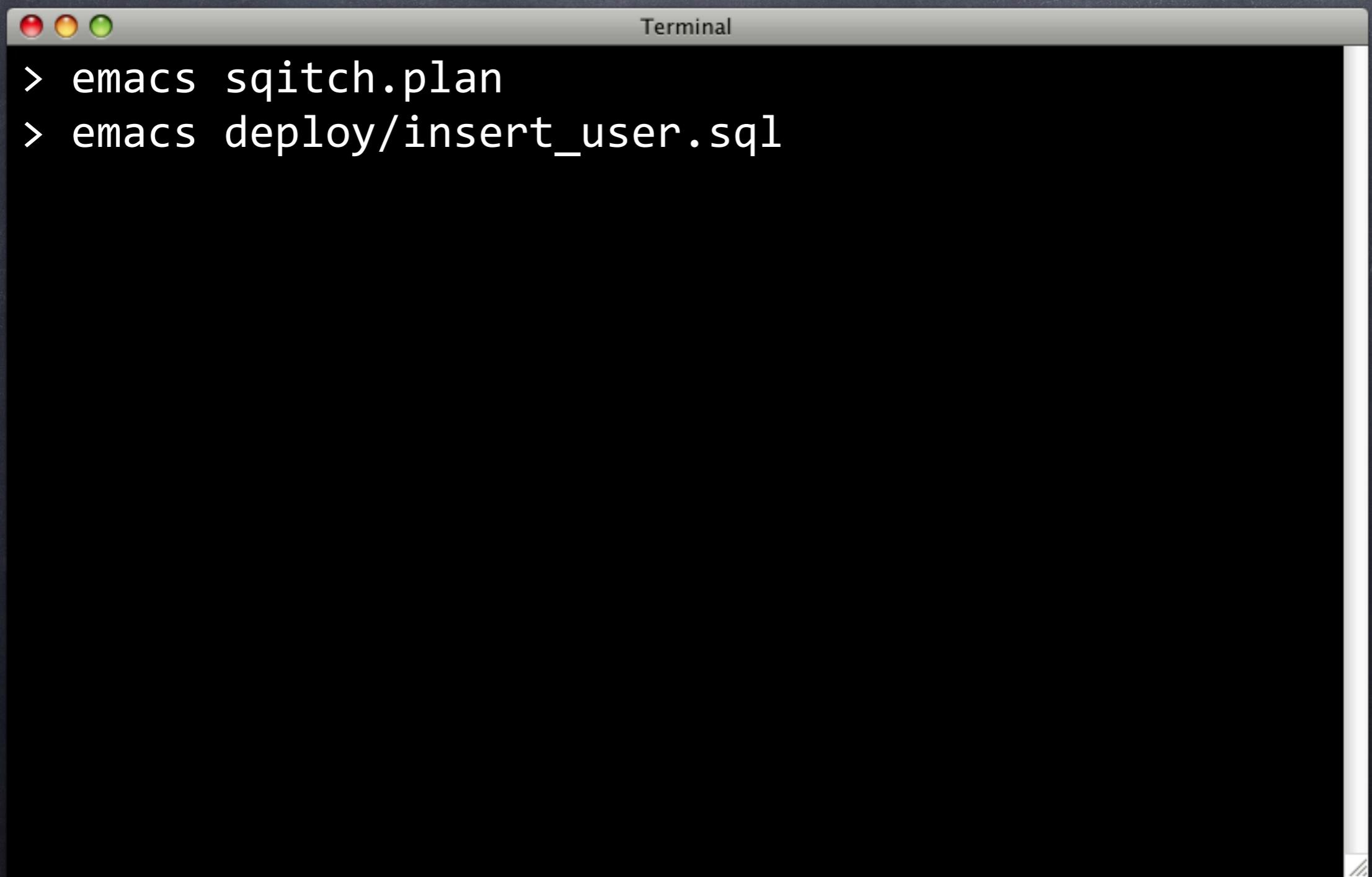
appschema 2012-08-17T01:49:33Z Marge N. O'Vera <marge@example.com> #
App schema for all flipr objects.
users [appschema] 2012-08-17T01:58:43Z Marge N. O'Vera
<marge@example.com> # Creates table to track our users.
insert_user [users appschema] 2012-08-17T02:07:53Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-08-17T02:08:10Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.

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

A Twofer

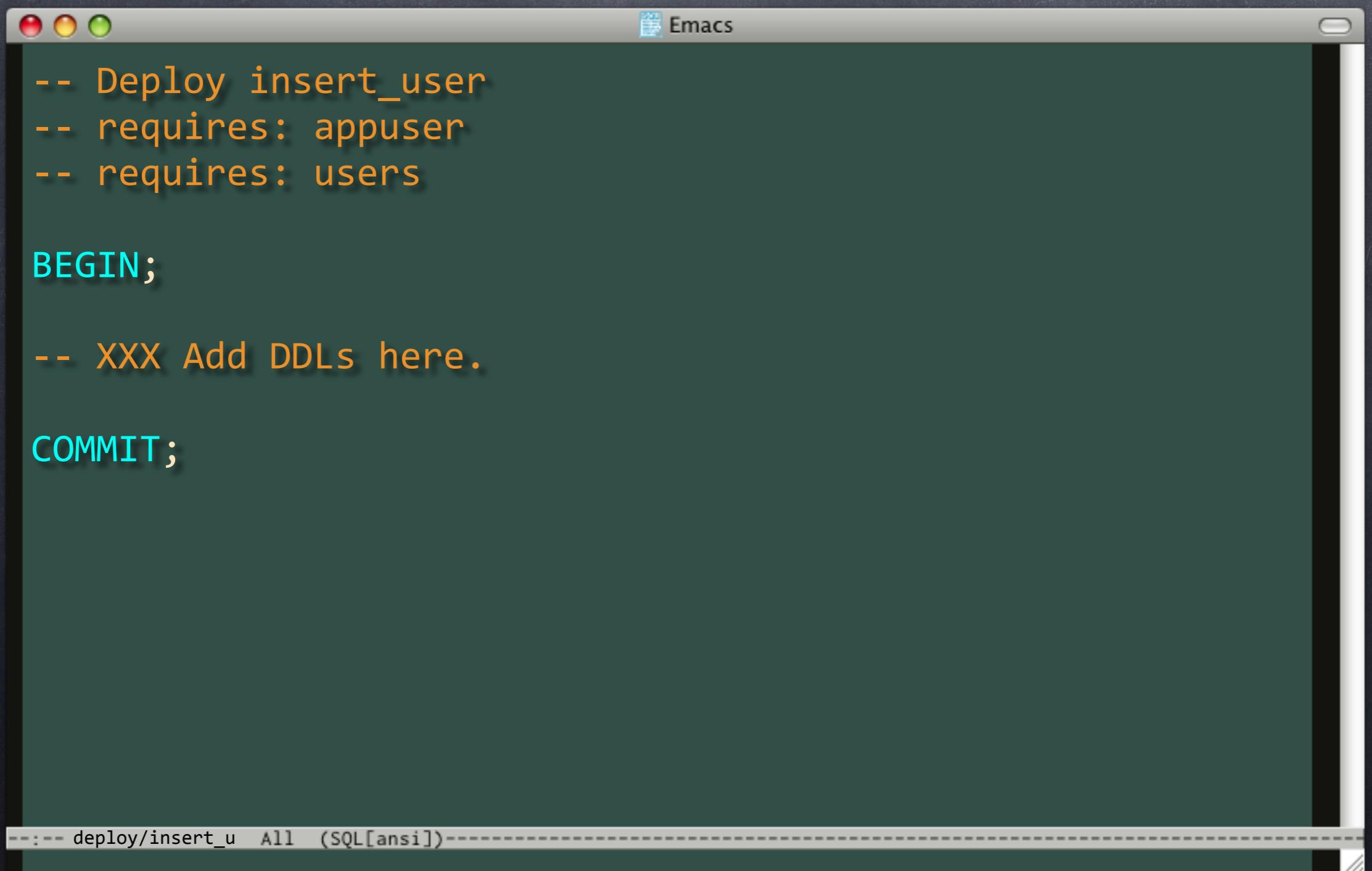


A Twofer



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

deploy/insert_user.sql



The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". The buffer contains the following SQL code:

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

BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

At the bottom of the window, there is a status bar with the text "---- deploy/insert_u All (SQL[ansi])-----".

deploy/insert_user.sql

The screenshot shows an Emacs window with a dark green background and white text. The title bar says "Emacs". The code is a SQL script:

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

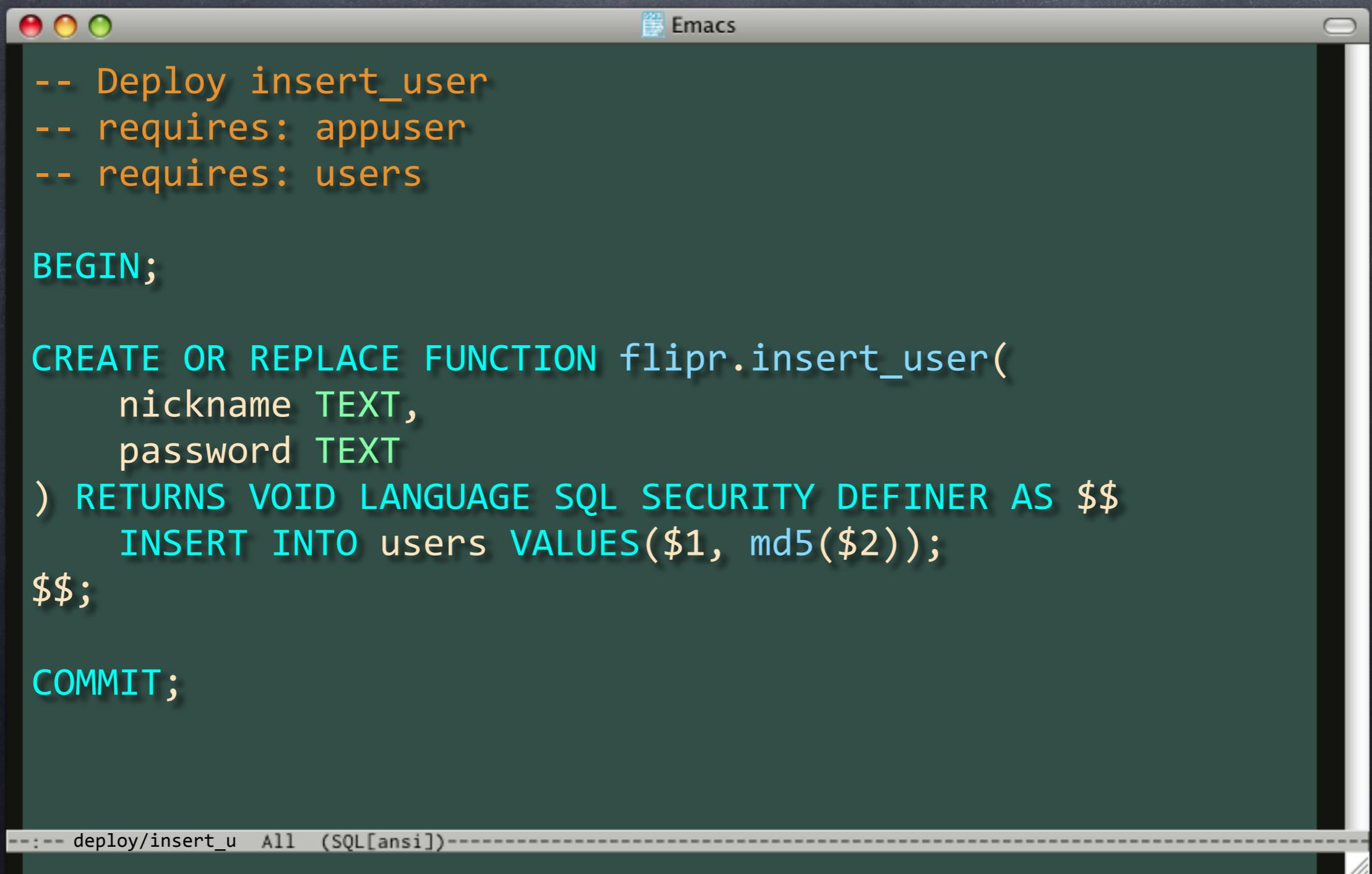
BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

At the bottom of the buffer, there is a status line with the text "---- deploy/insert_u All (SQL[ansi])-----".

deploy/insert_user.sql



The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". The code inside the window is as follows:

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

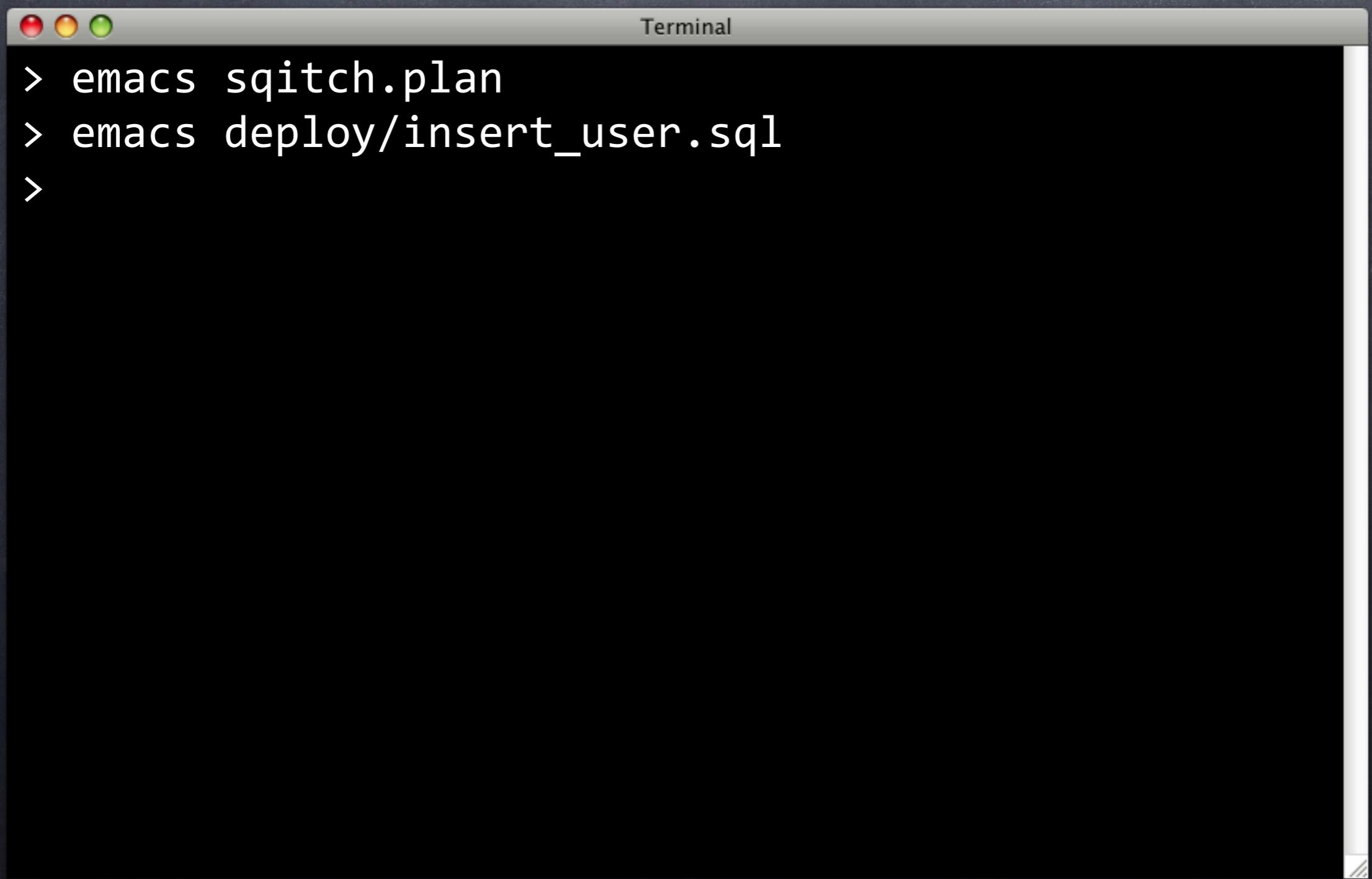
BEGIN;

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;

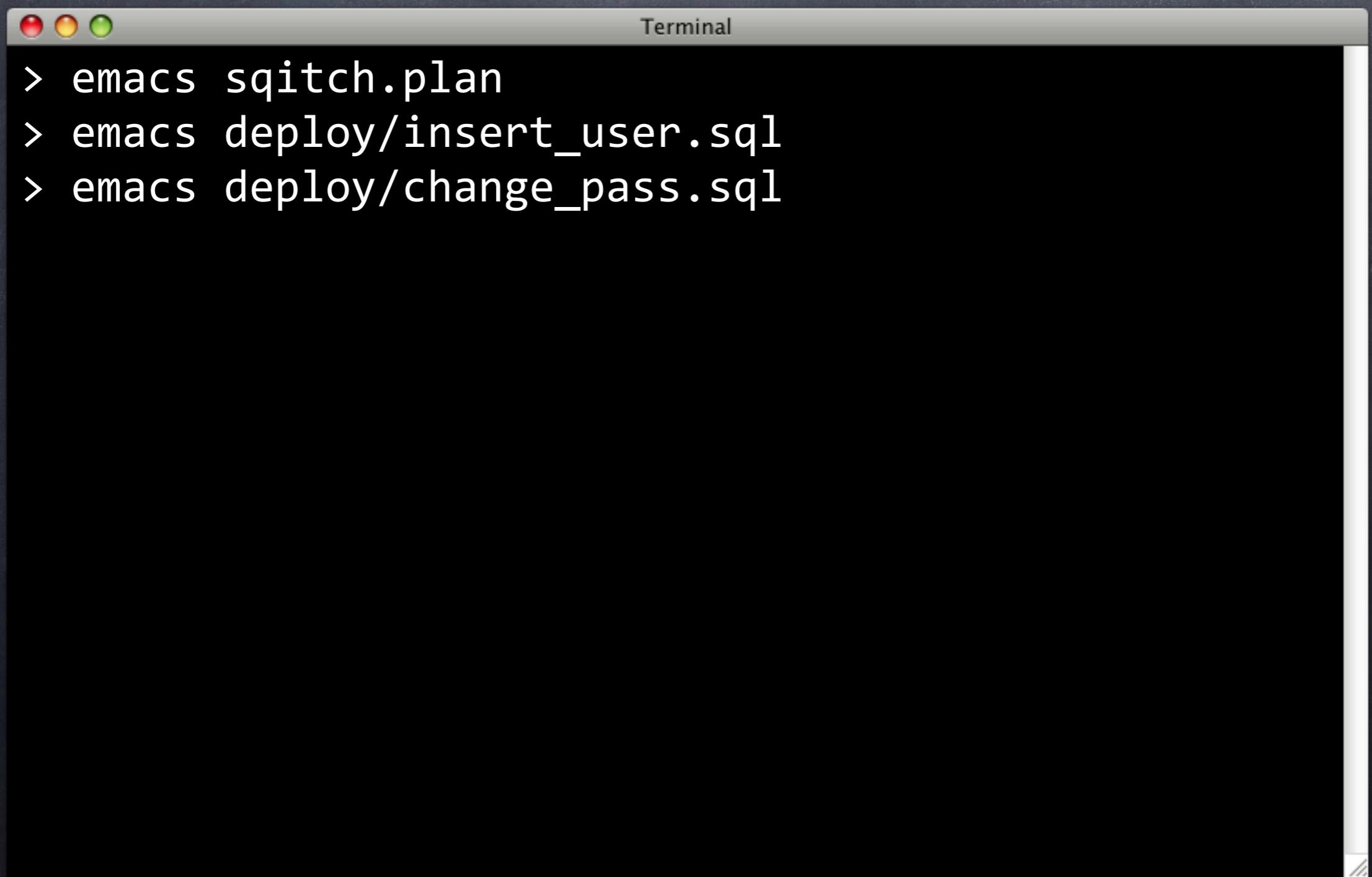
---- deploy/insert_u  All  (SQL[ansi])----
```

A Twofer



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

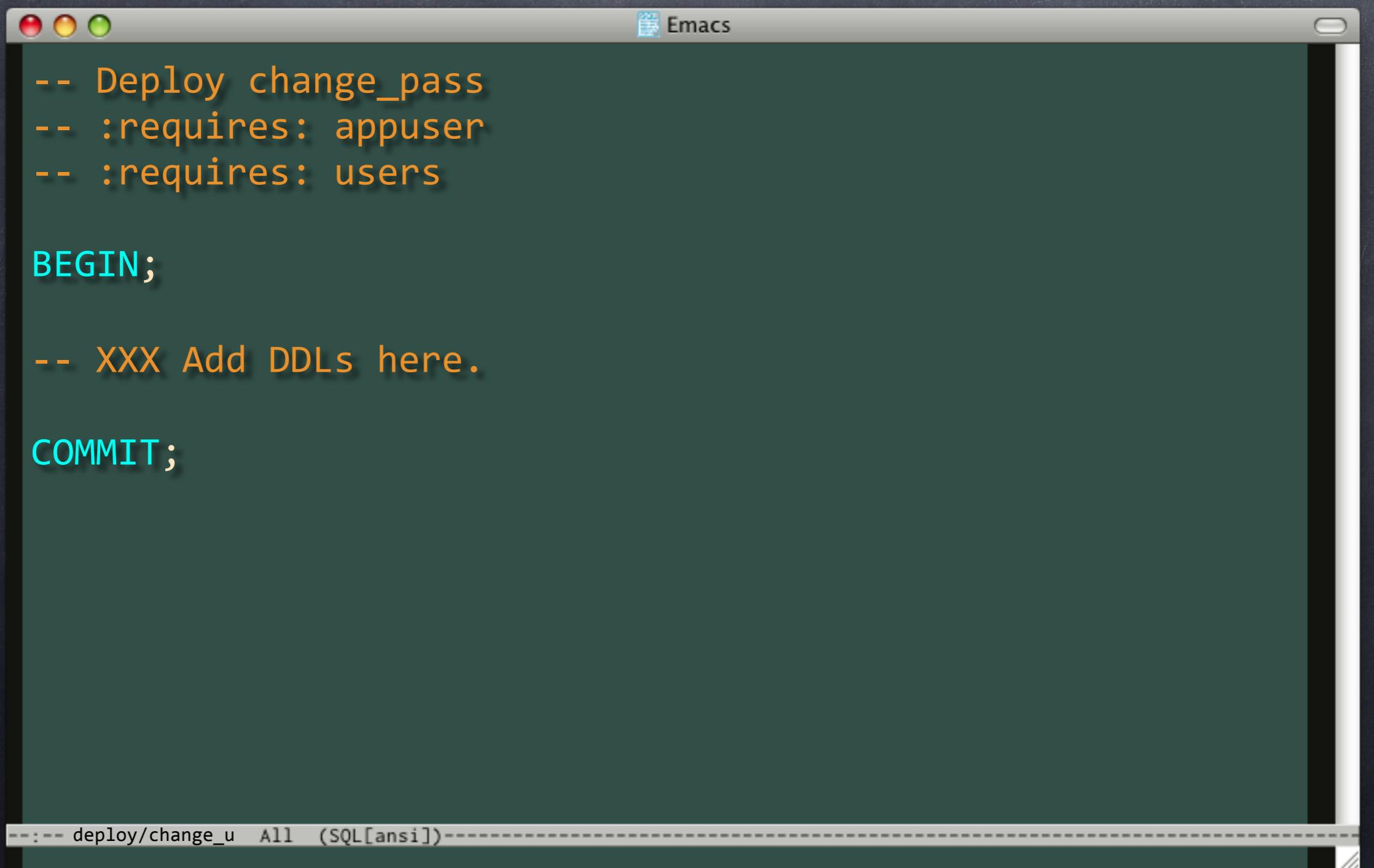
A Twofer



The image shows a screenshot of a Mac OS X Terminal window. The window has a title bar with three colored buttons (red, yellow, green) on the left and the word "Terminal" centered above a dark gray content area. Inside the terminal, there are three white text lines, each starting with a right-pointing arrowhead (">") followed by the command "emacs" and a file path. The paths are "sqitch.plan", "deploy/insert_user.sql", and "deploy/change_pass.sql".

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

deploy/change_pass.sql



The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". The buffer contains the following SQL code:

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

BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

In the bottom status bar, there is a message: "---- deploy/change_u All (SQL[ansi])----".

deploy/change_pass.sql

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

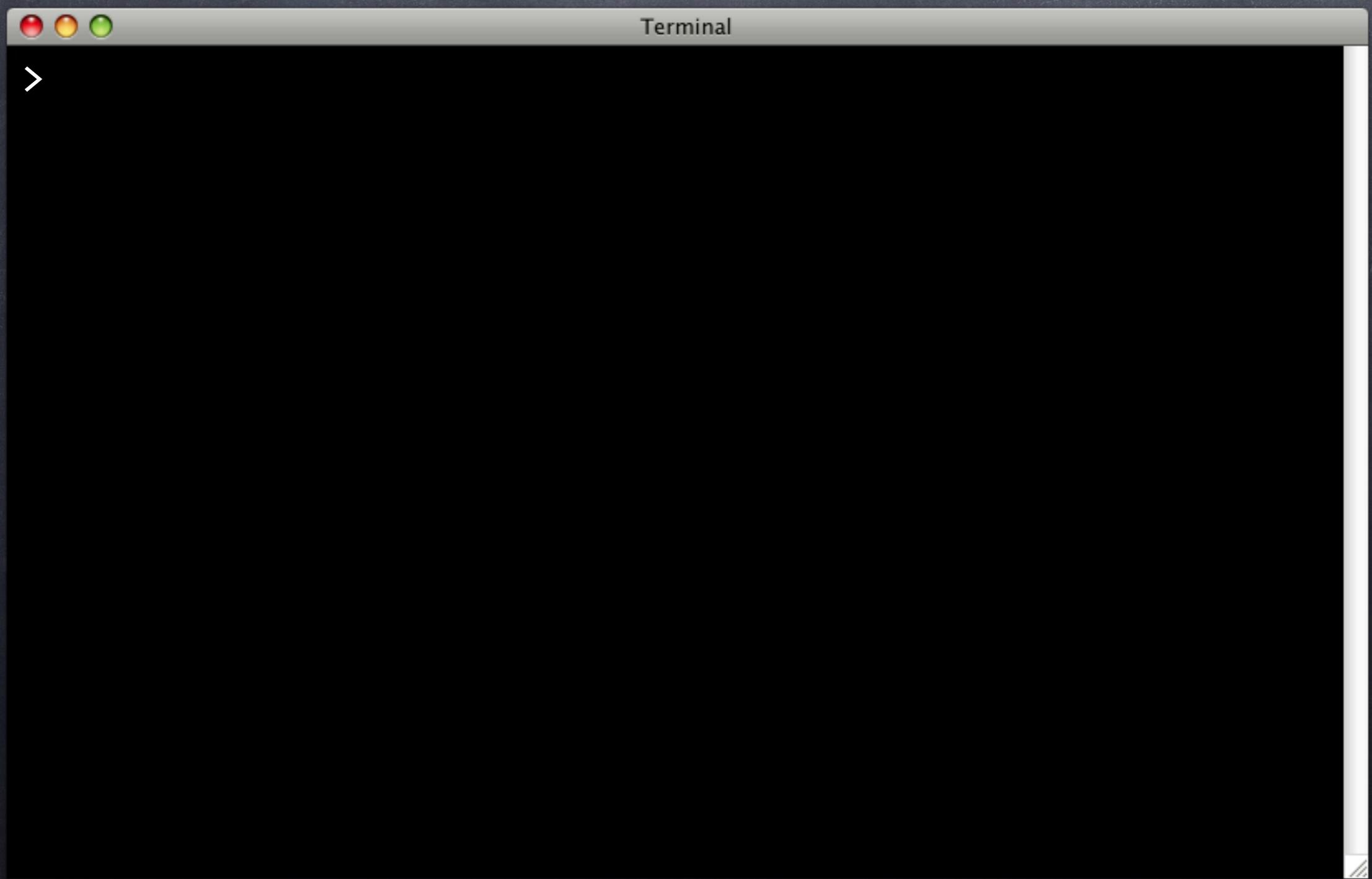
BEGIN;
CREATE OR REPLACE FUNCTION flipr.change_pass(
    nick TEXT, oldpass TEXT, newpass TEXT
) RETURNS BOOLEAN LANGUAGE plpgsql SECURITY DEFINER AS $$

BEGIN
    UPDATE users SET password = md5($3)
        WHERE nickname = $1 AND password = md5($2);
    RETURN FOUND;
END;
$$;

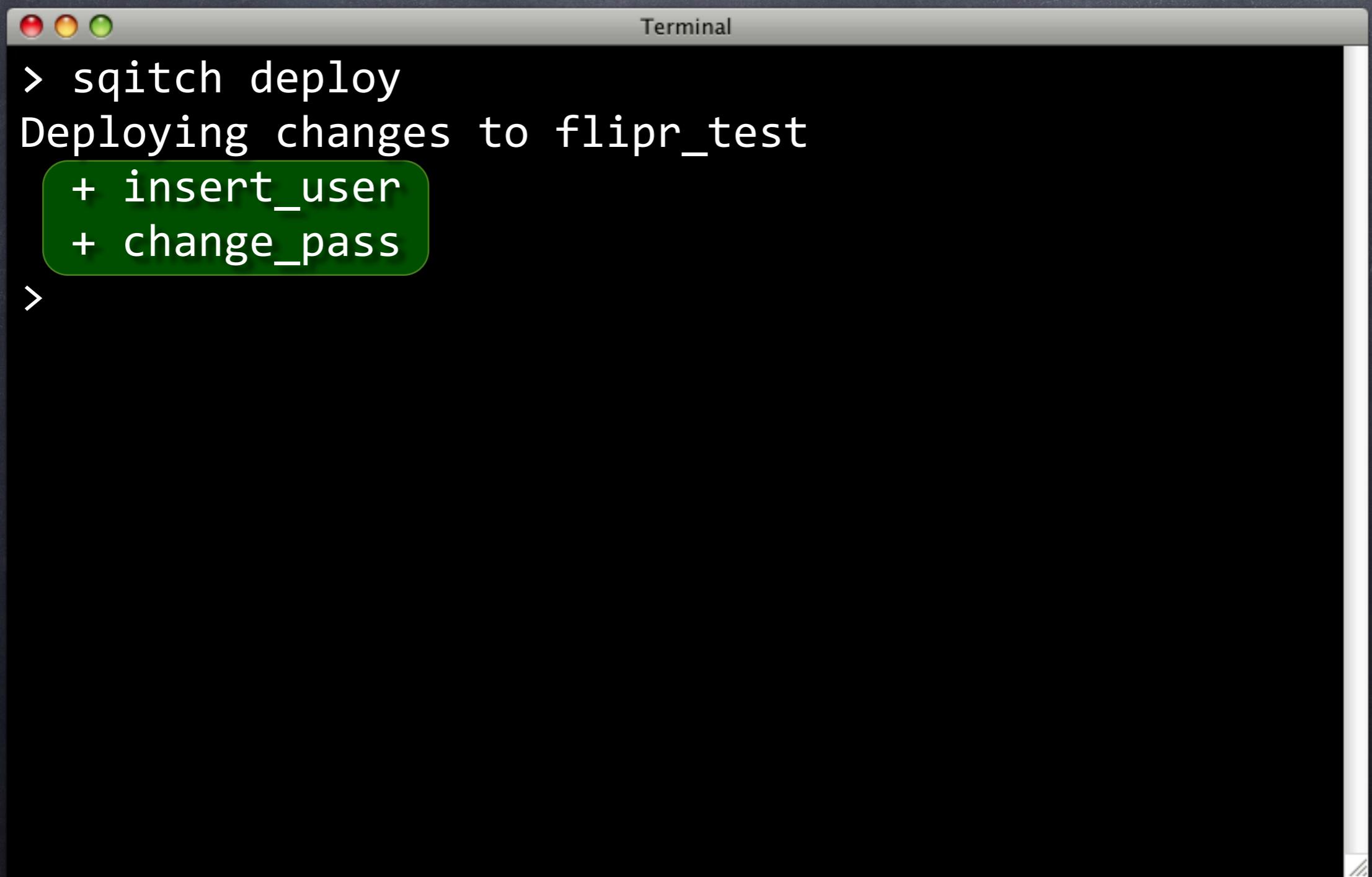
COMMIT;

--- deploy/change_u  All  (SQL[ansi])---
```

Deploy Functions



Deploy Functions



```
Terminal
> sqitch deploy
Deploying changes to flipr_test
+ insert_user
+ change_pass
>
```

A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the command "sqitch deploy" followed by the output of the deployment process. The output shows "Deploying changes to flipr_test" and two function names: "insert_user" and "change_pass". A green rounded rectangle highlights these two function names. The terminal has its standard red, yellow, and green close buttons at the top left.

Deploy Functions

```
Terminal
> sqitch deploy
Deploying changes to flipr_test
  + insert_user
  + change_pass
> 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
>
```

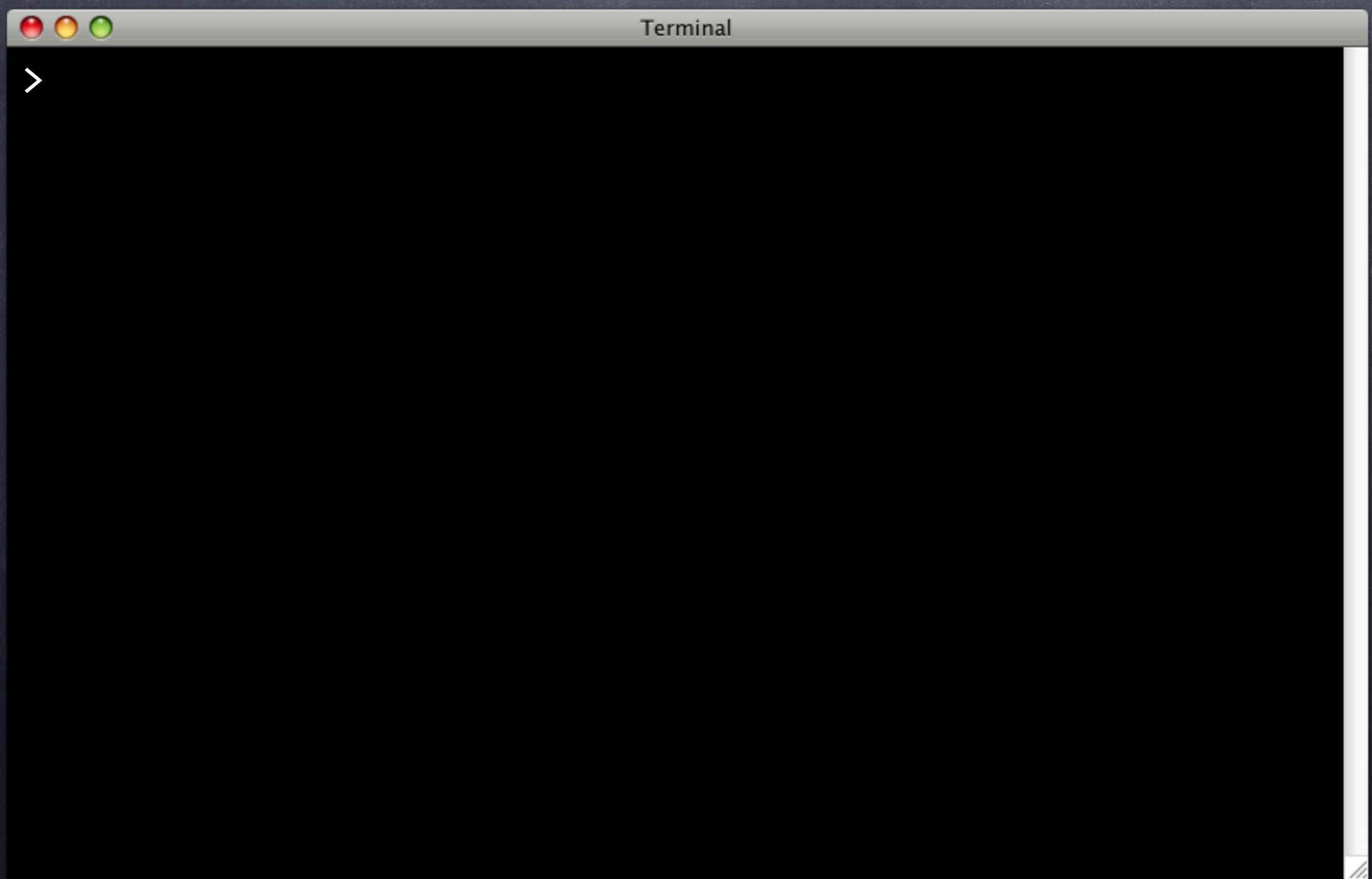
Deploy Functions

```
Terminal  
> sqitch deploy  
Deploying changes to flipr_test  
  + insert_user  
  + change_pass  
> psql -d flipr_test -c '\df flipr.*'  
                                         List of function  

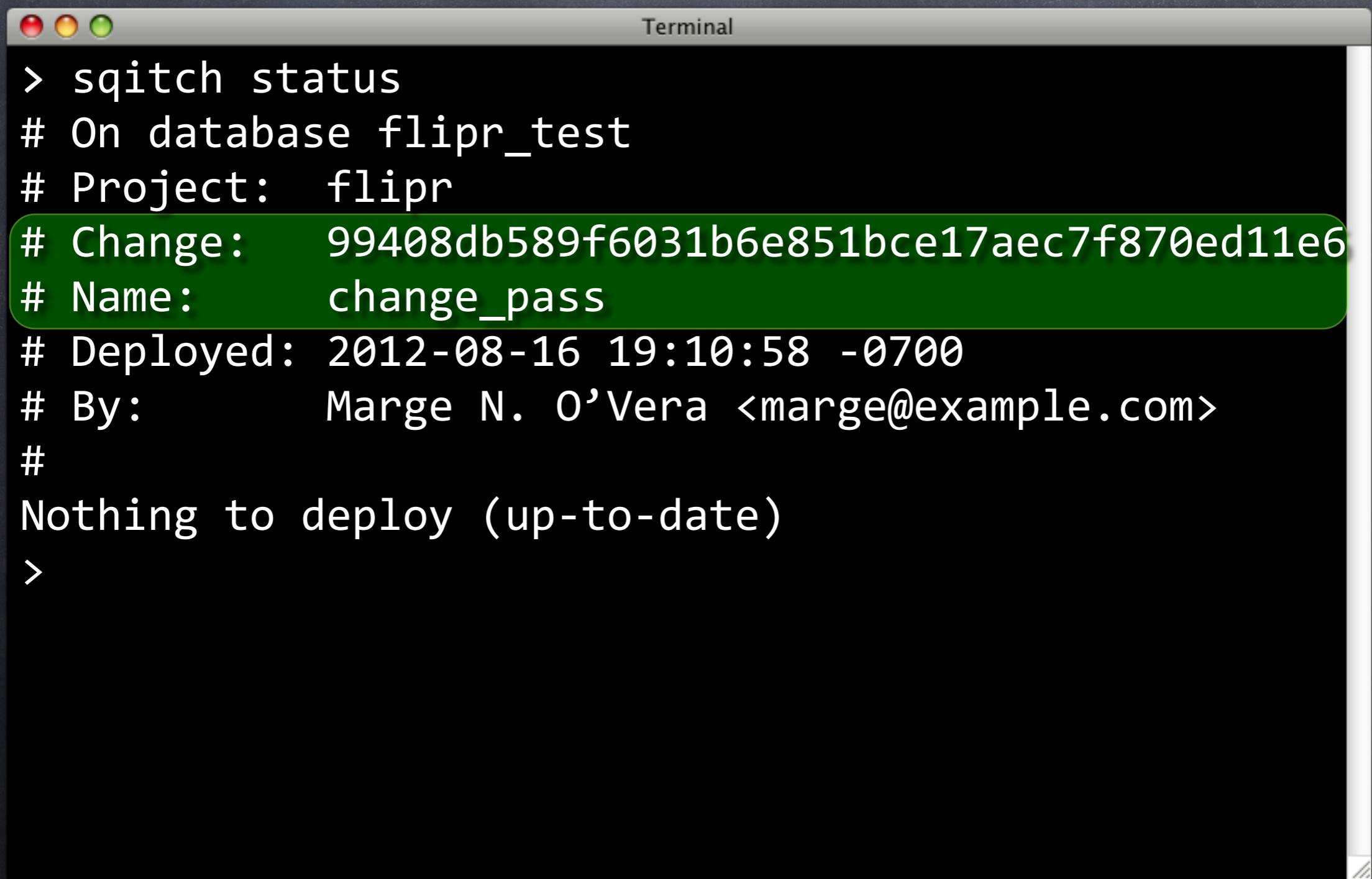

| Schema | Name        | Result data type |           |
|--------|-------------|------------------|-----------|
| flipr  | change_pass | boolean          | nick text |
| flipr  | insert_user | void             | nickname  |

  
>
```

Status Update

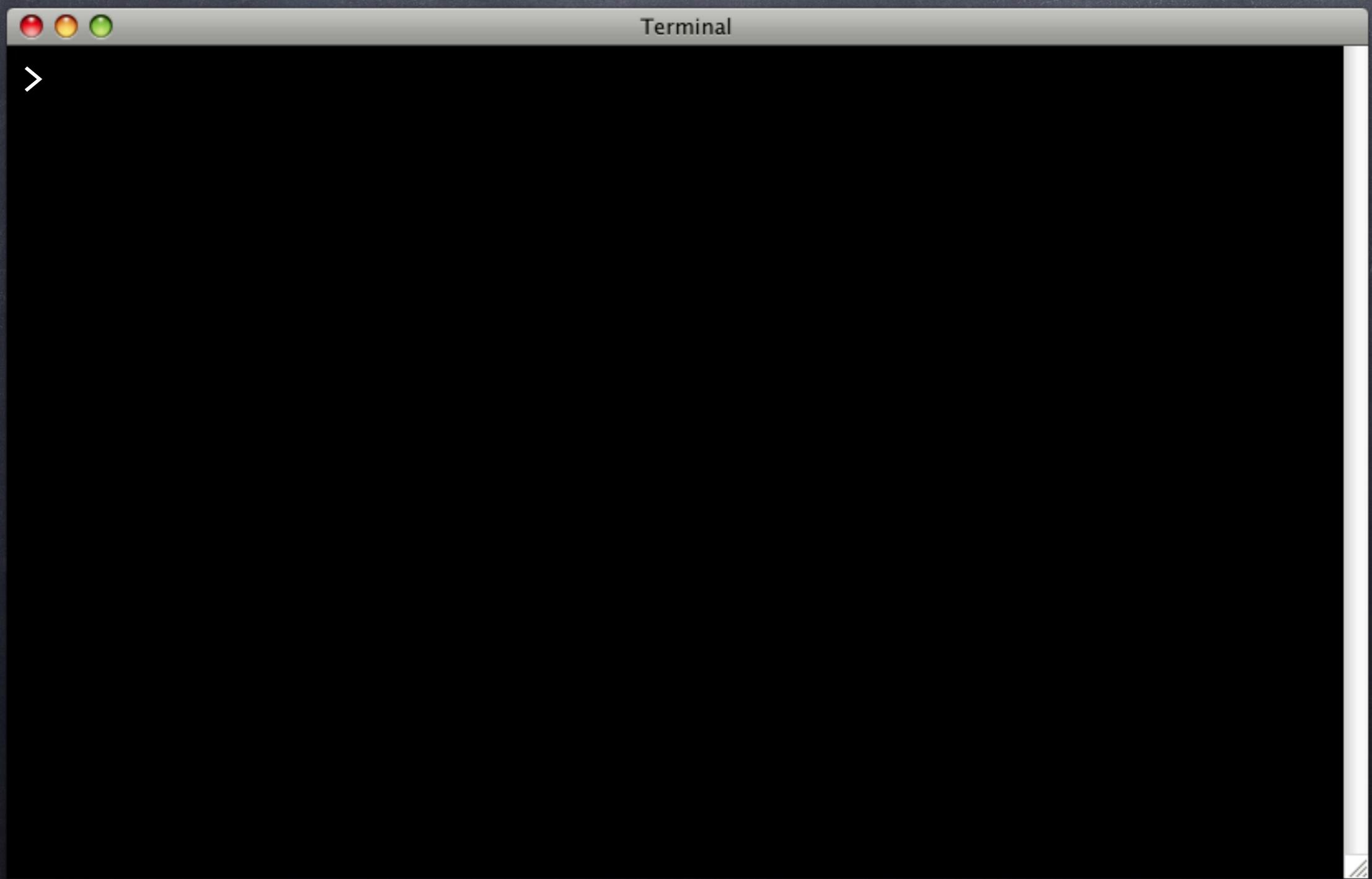


Status Update

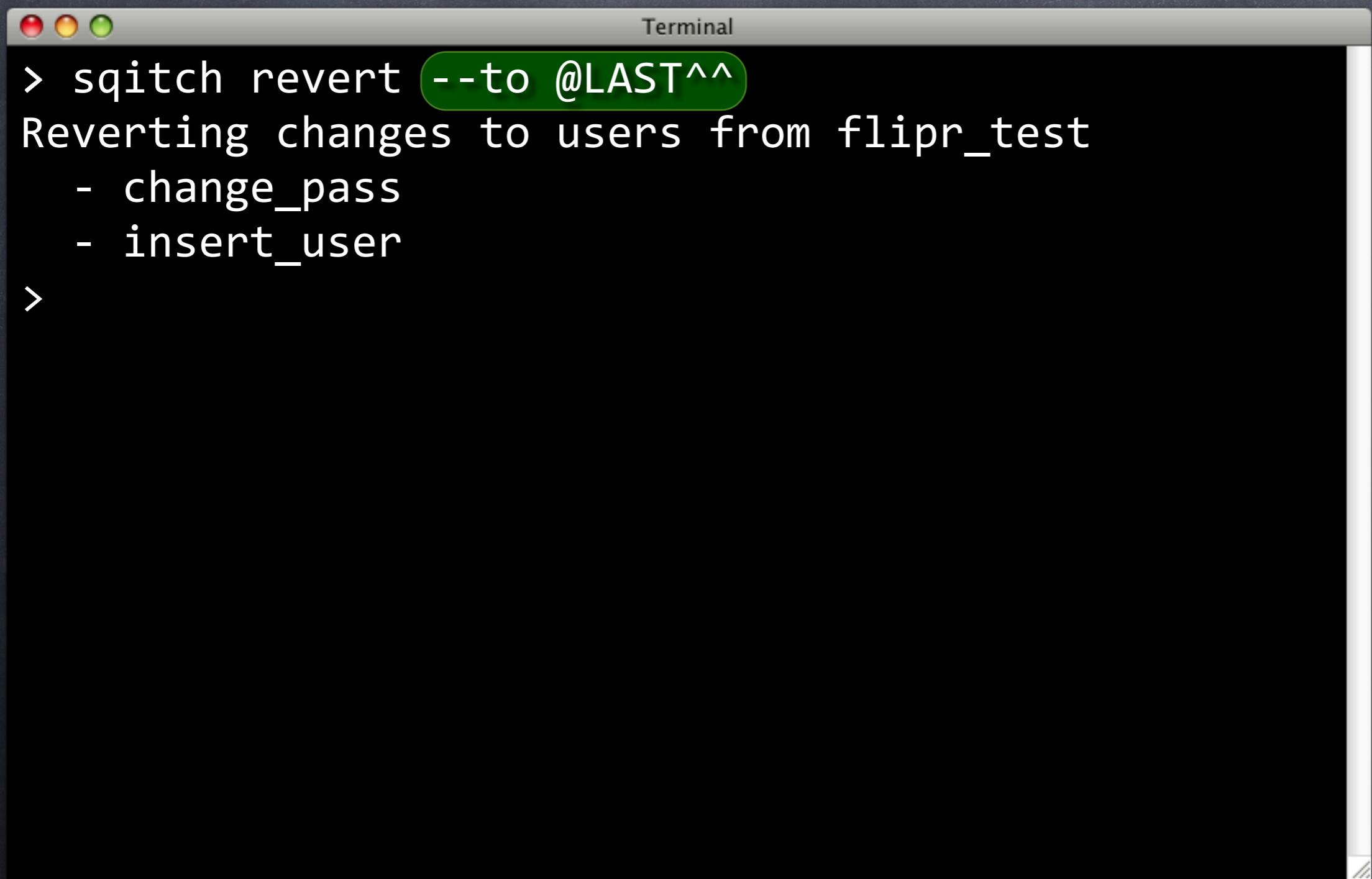
A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the output of the command "sqitch status". The output shows details about a database named "flipr_test" under a project "flipr". A specific change set is highlighted with a green rounded rectangle, containing the ID "99408db589f6031b6e851bce17aec7f870ed11e6", a name "change_pass", a deployment date and time "2012-08-16 19:10:58 -0700", and a deployer "Marge N. O'Vera <marge@example.com>". Below this, a note says "Nothing to deploy (up-to-date)".>
sqitch status
On database flipr_test
Project: flipr
Change: 99408db589f6031b6e851bce17aec7f870ed11e6
Name: change_pass
Deployed: 2012-08-16 19:10:58 -0700
By: Marge N. O'Vera <marge@example.com>

Nothing to deploy (up-to-date)
>

Revert



Revert



A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The main area contains a command-line session:

```
> sqitch revert --to @LAST^^
Reverting changes to users from flipr_test
- change_pass
- insert_user
>
```

The command `--to @LAST^^` is highlighted with a green rounded rectangle.

Revert

```
Terminal  
> sqitch revert --to @LAST^^  
Reverting changes to users from flipr_test  
  - change_pass  
  - insert_user  
>
```

Revert

```
Terminal  
> sqitch revert --to @LAST^^  
Reverting changes to users from flipr_test  
- change_pass  
- insert_user  
> 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

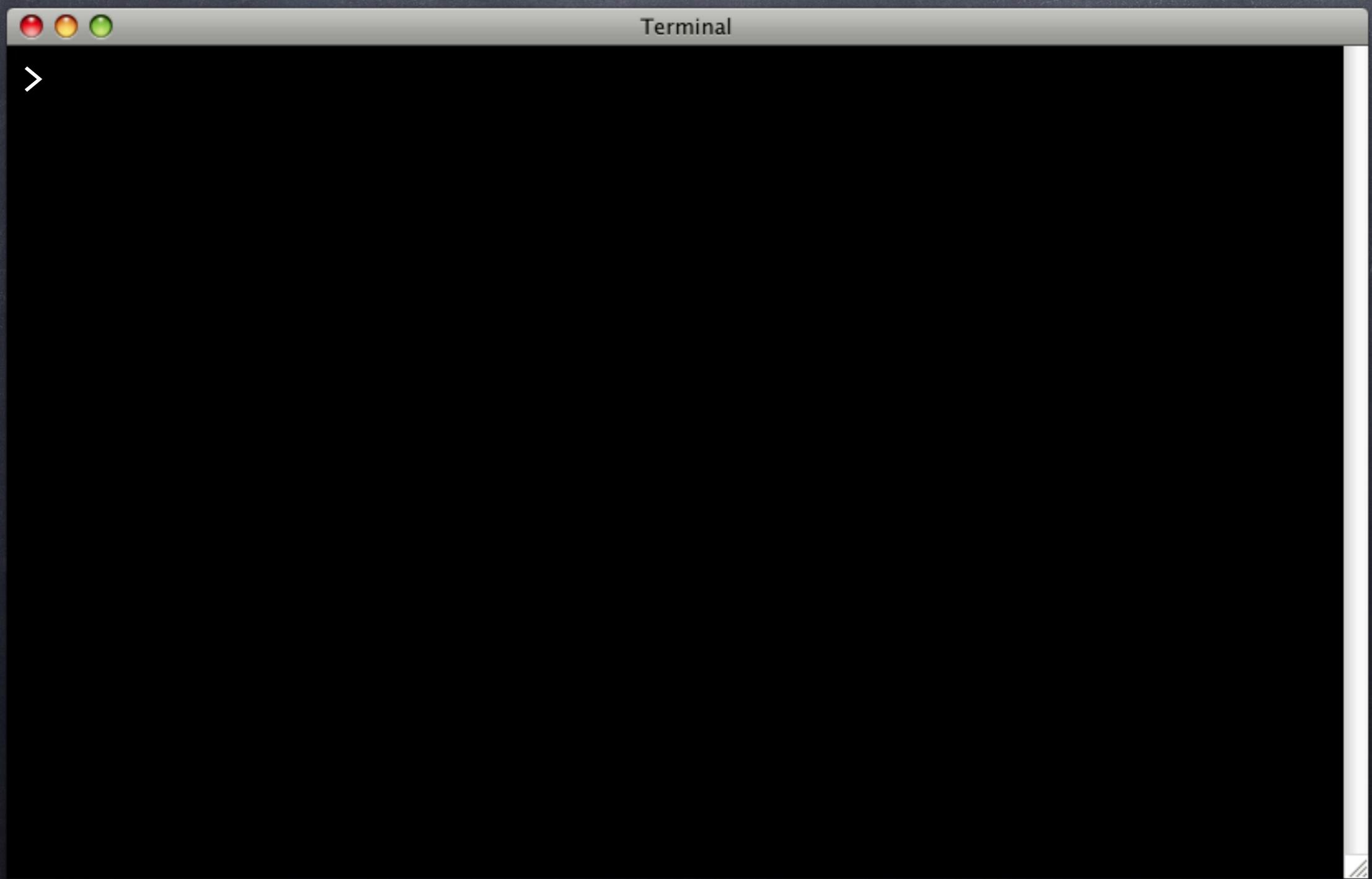
Commit It

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

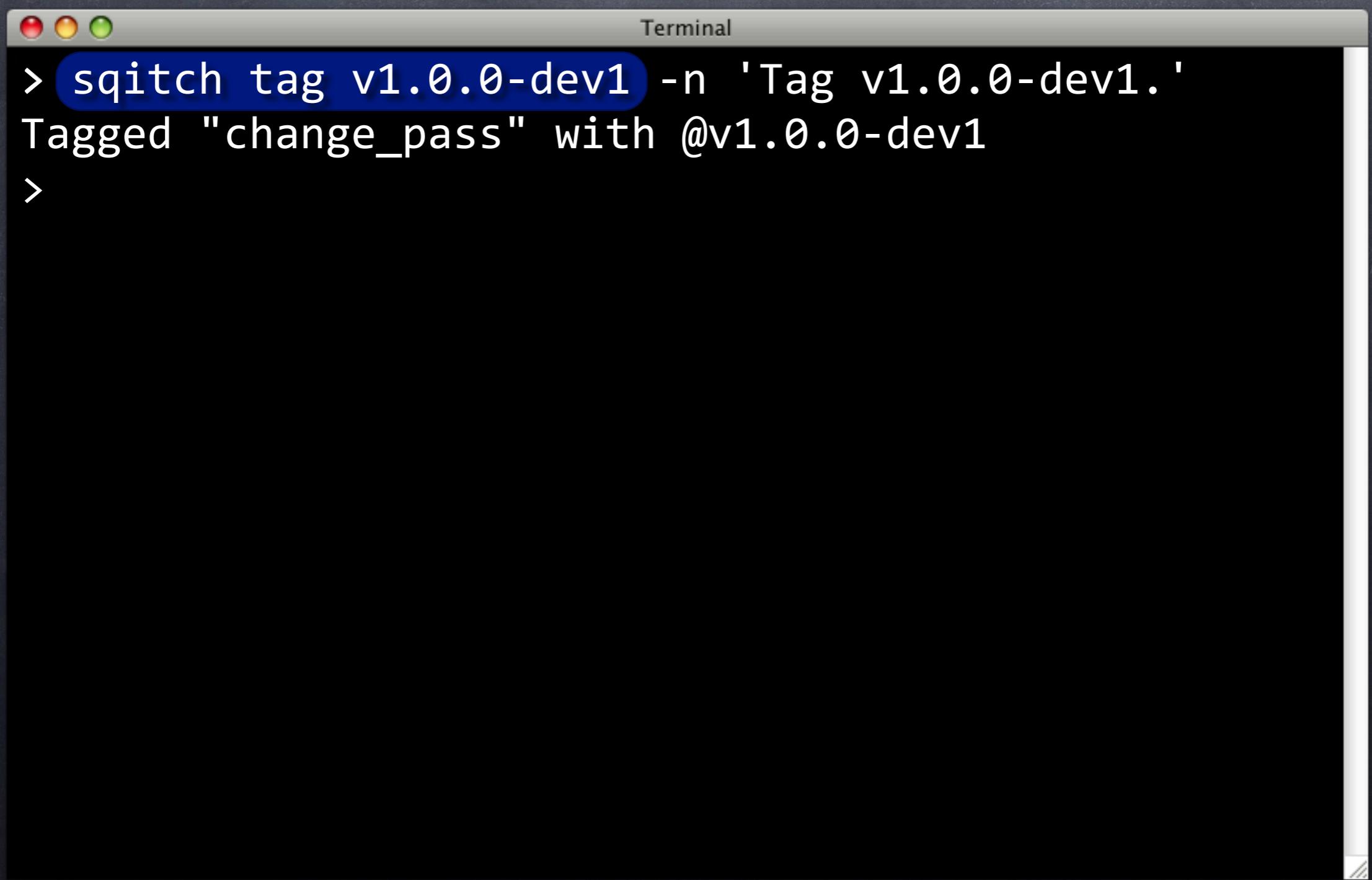
Commit It

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

Tag It!

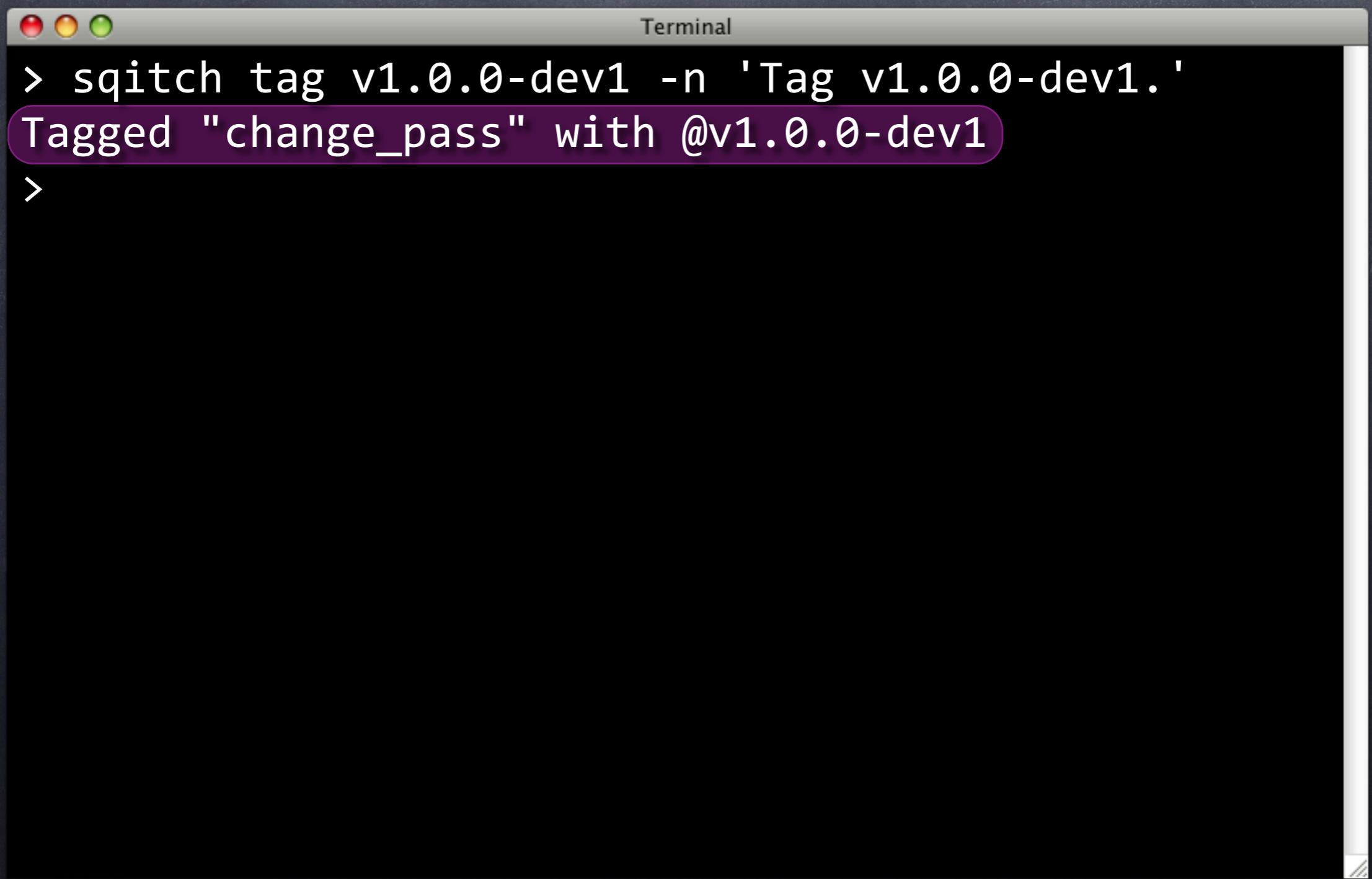


Tag It!



```
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  
>
```

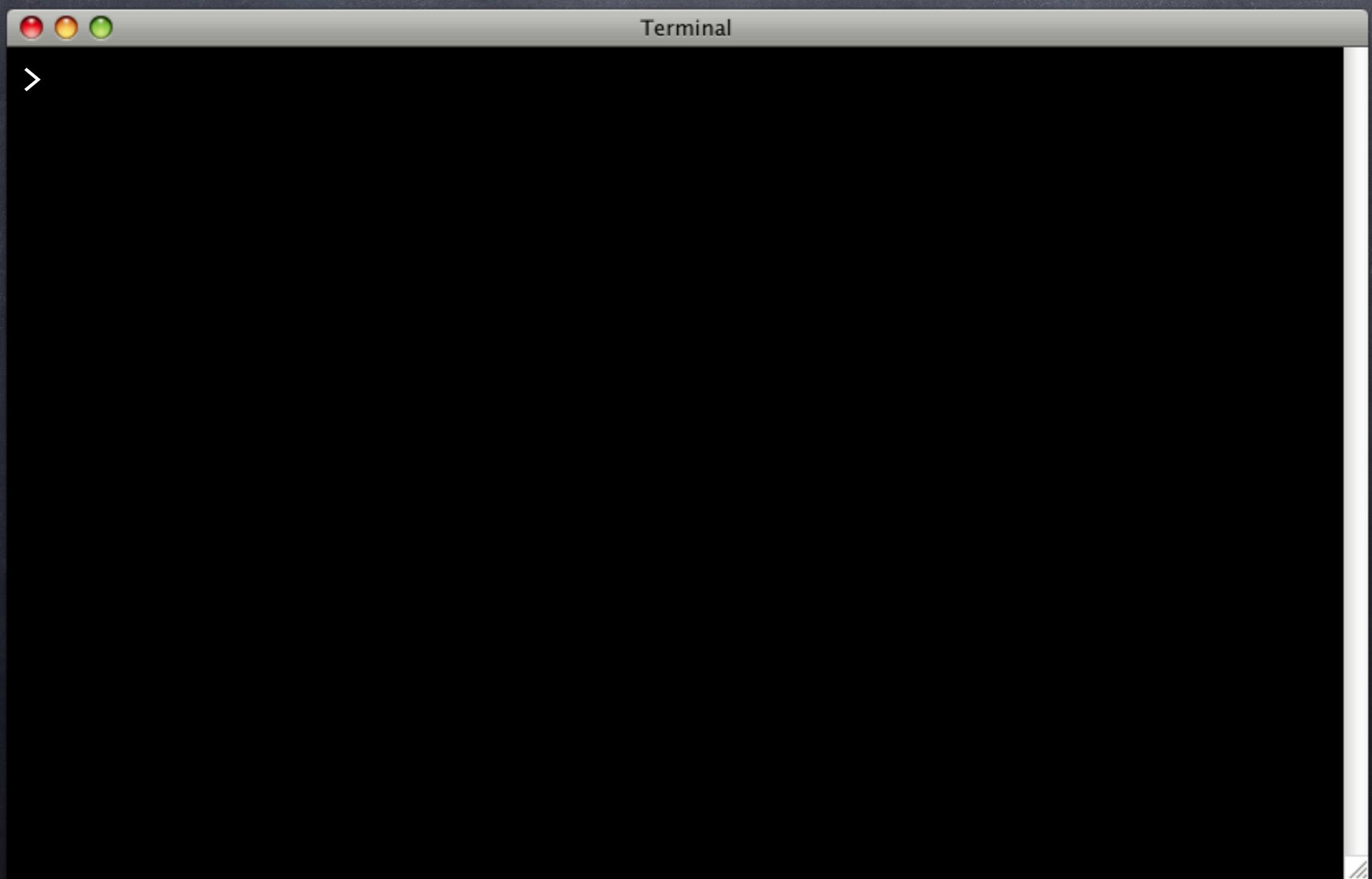
Tag It!

```
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 067ef73] Tag the database with v1.0.0-dev1.  
 1 file changed, 1 insertion(+)  
>
```

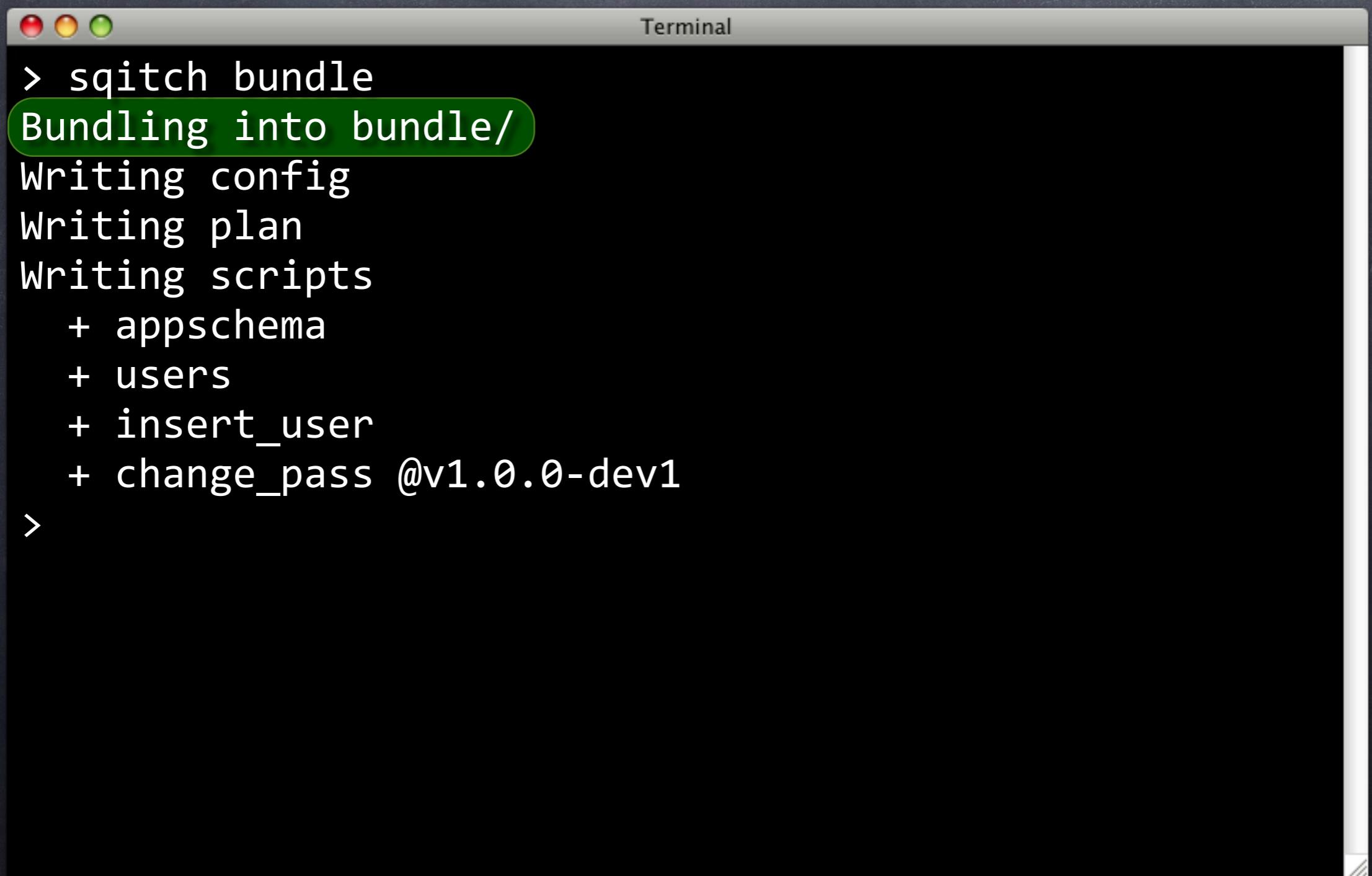
Tag It!

```
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 067ef73] 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!



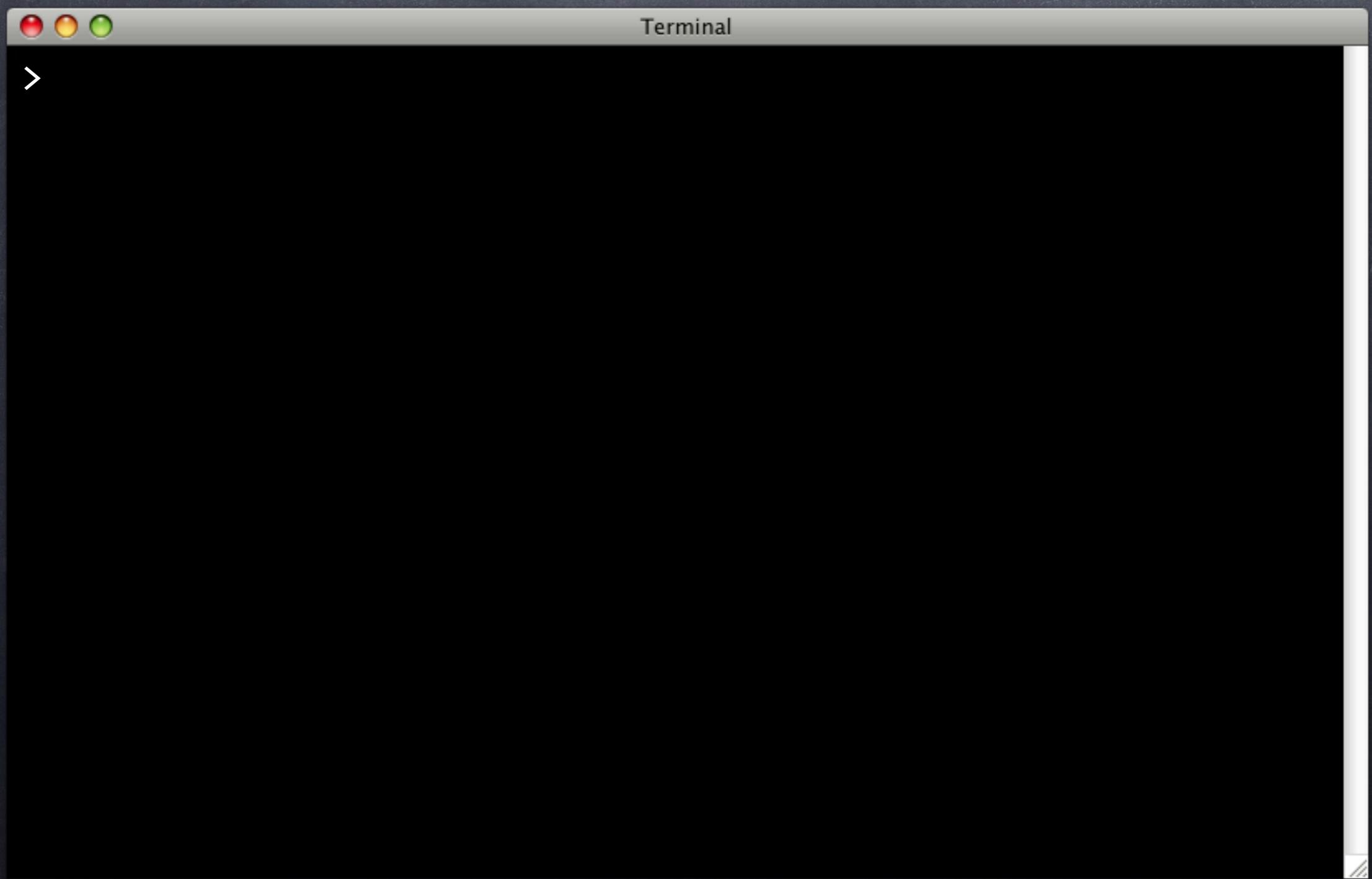
A screenshot of a Mac OS X Terminal window titled "Terminal". The window shows the command "sqitch bundle" being run, followed by the output of the command. The output includes the message "Bundling into bundle/", followed by "Writing config", "Writing plan", "Writing scripts", and a list of generated files: "+ appschema", "+ users", "+ insert_user", and "+ change_pass @v1.0.0-dev1". The entire output is displayed in white text on a black background, with the "Bundling into bundle/" message highlighted with a green rounded rectangle.

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

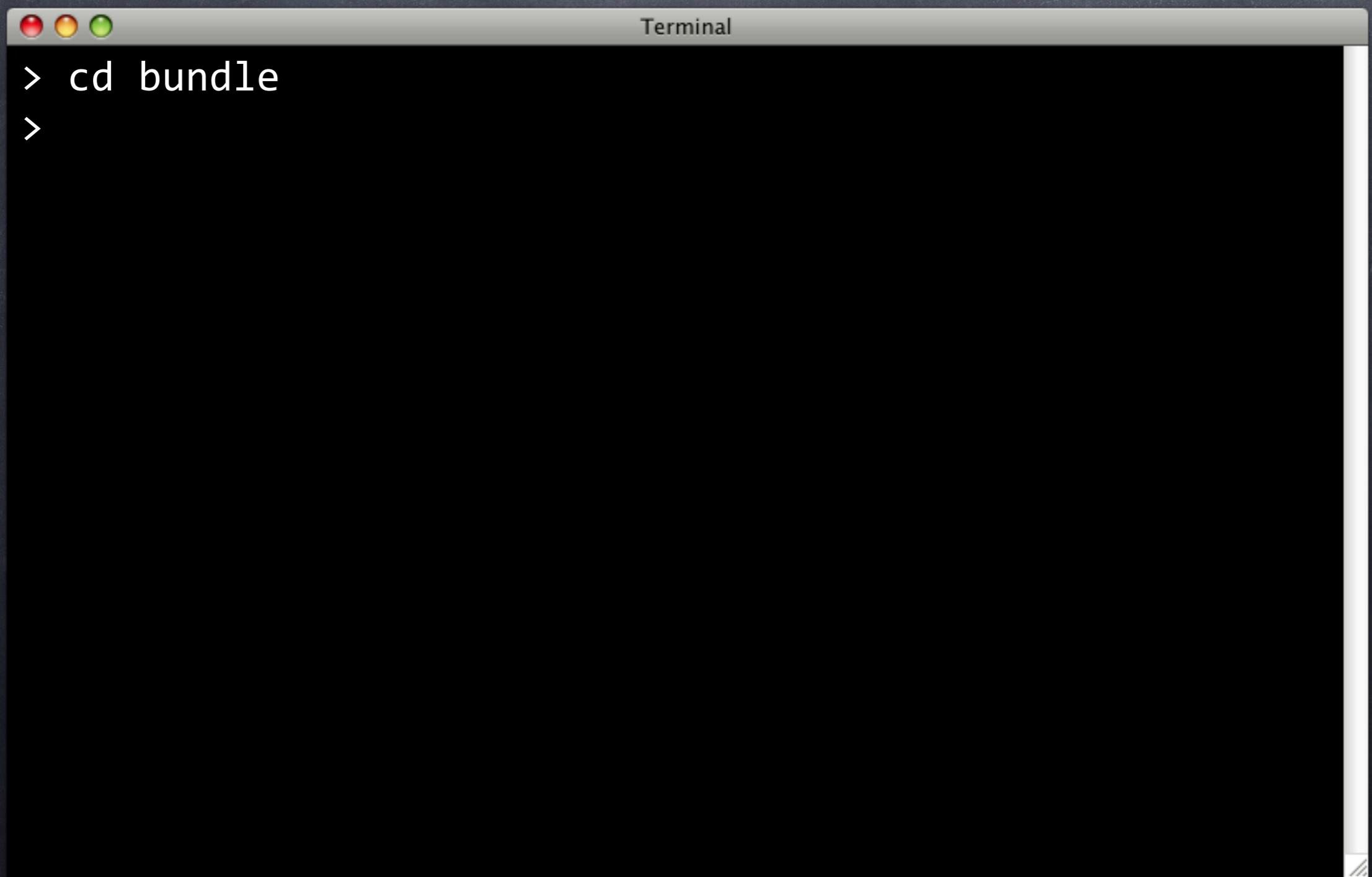
Bundle It!

```
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!

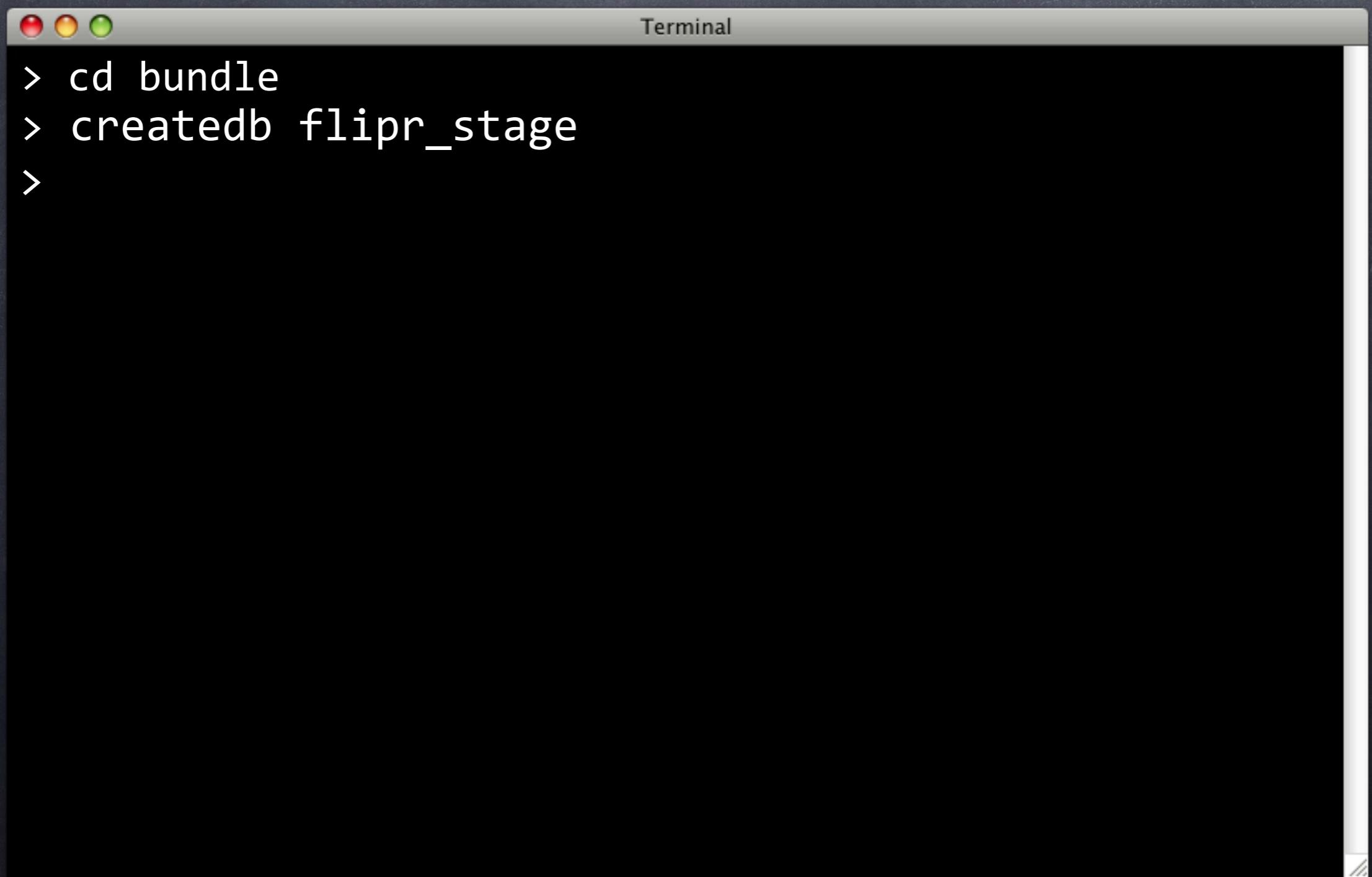


Test and Ship It!



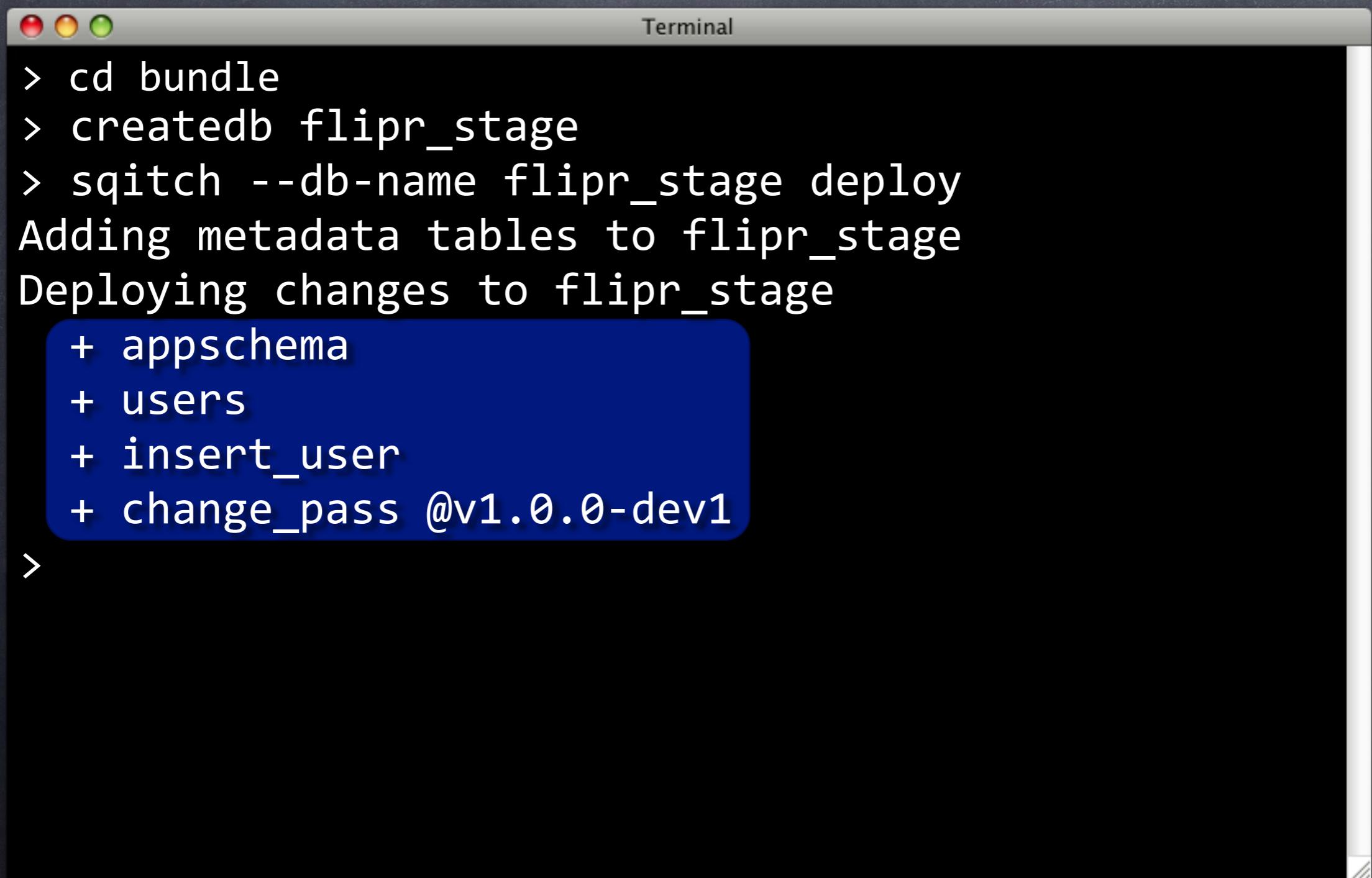
```
Terminal
> cd bundle
>
```

Test and Ship It!



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

Test and Ship It!



A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the following text:

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

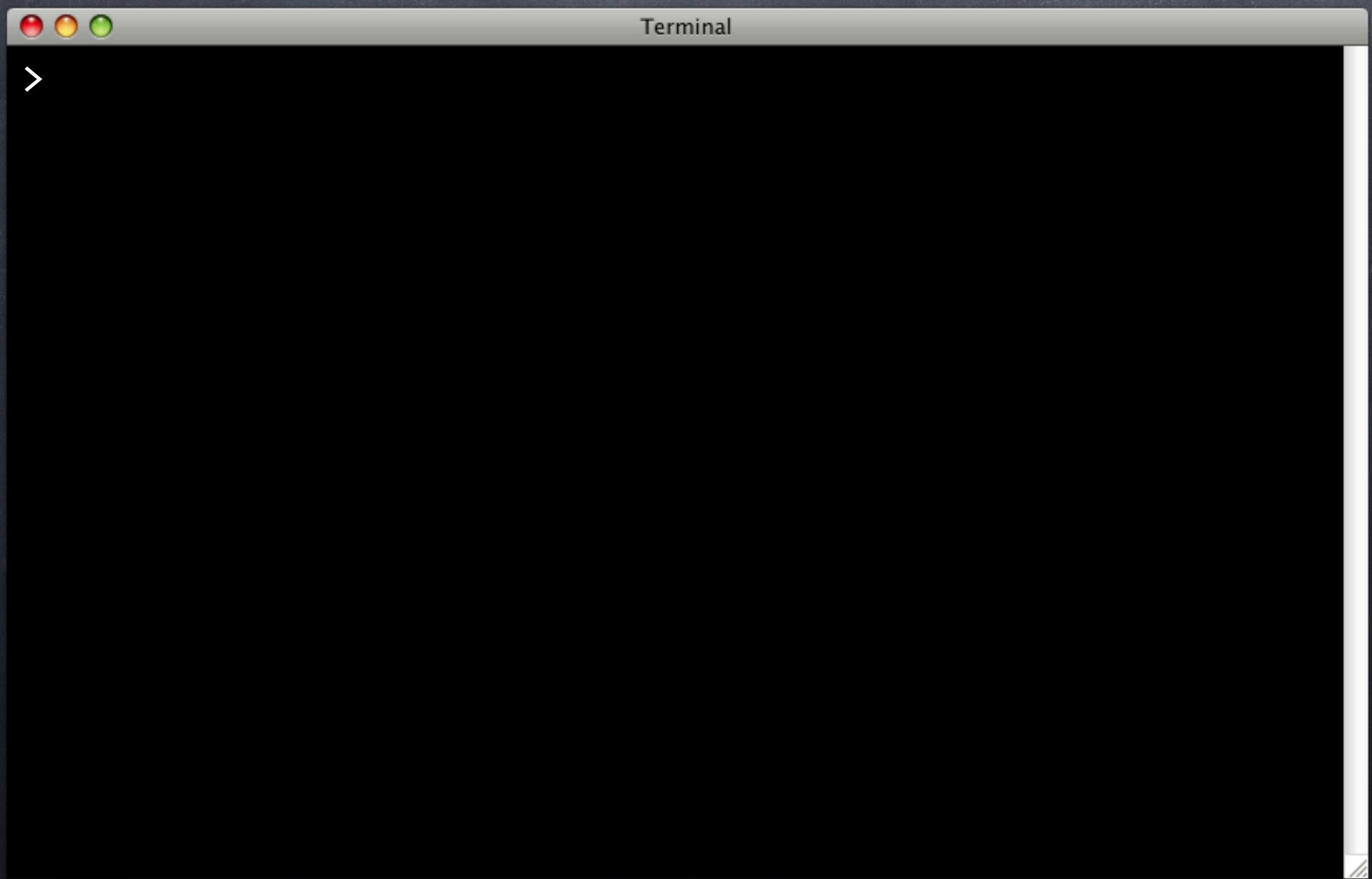
The last four lines of the output are highlighted with a blue rounded rectangle.

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  
+ users  
+ insert_user  
+ change_pass @v1.0.0-dev1  
> 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

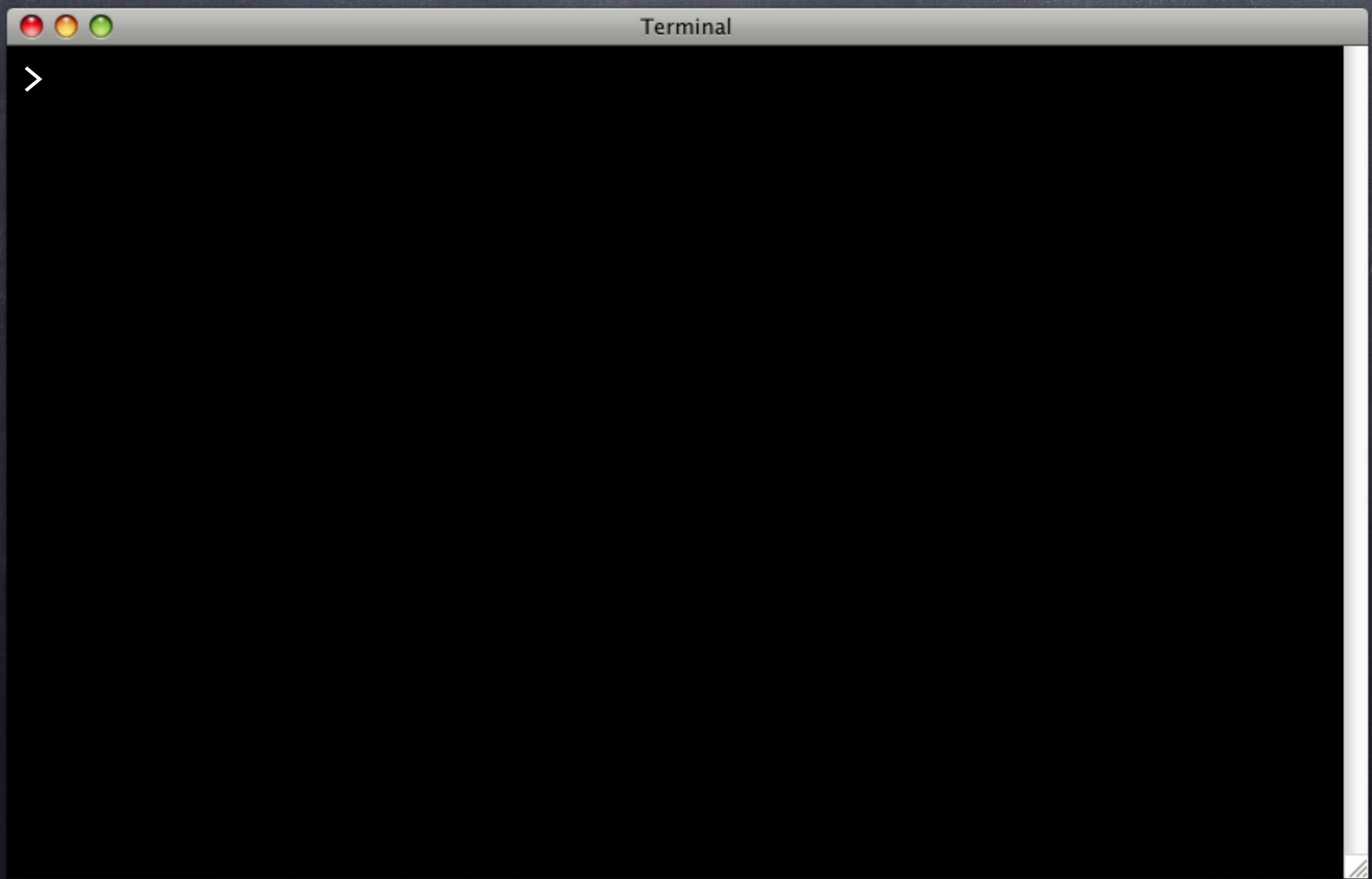
```
Terminal  
> psql -d flipr_test -c "  
    SELECT insert_user('foo', 'secr3t'),  
          insert_user('bar', 'secr3t');  
    SELECT nickname, password FROM users;  
"  
  
nickname | password  
-----+-----  
foo      | 9695da4dd567a19f9b92065f240c6725  
bar      | 9695da4dd567a19f9b92065f240c6725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

Ruh-Roh

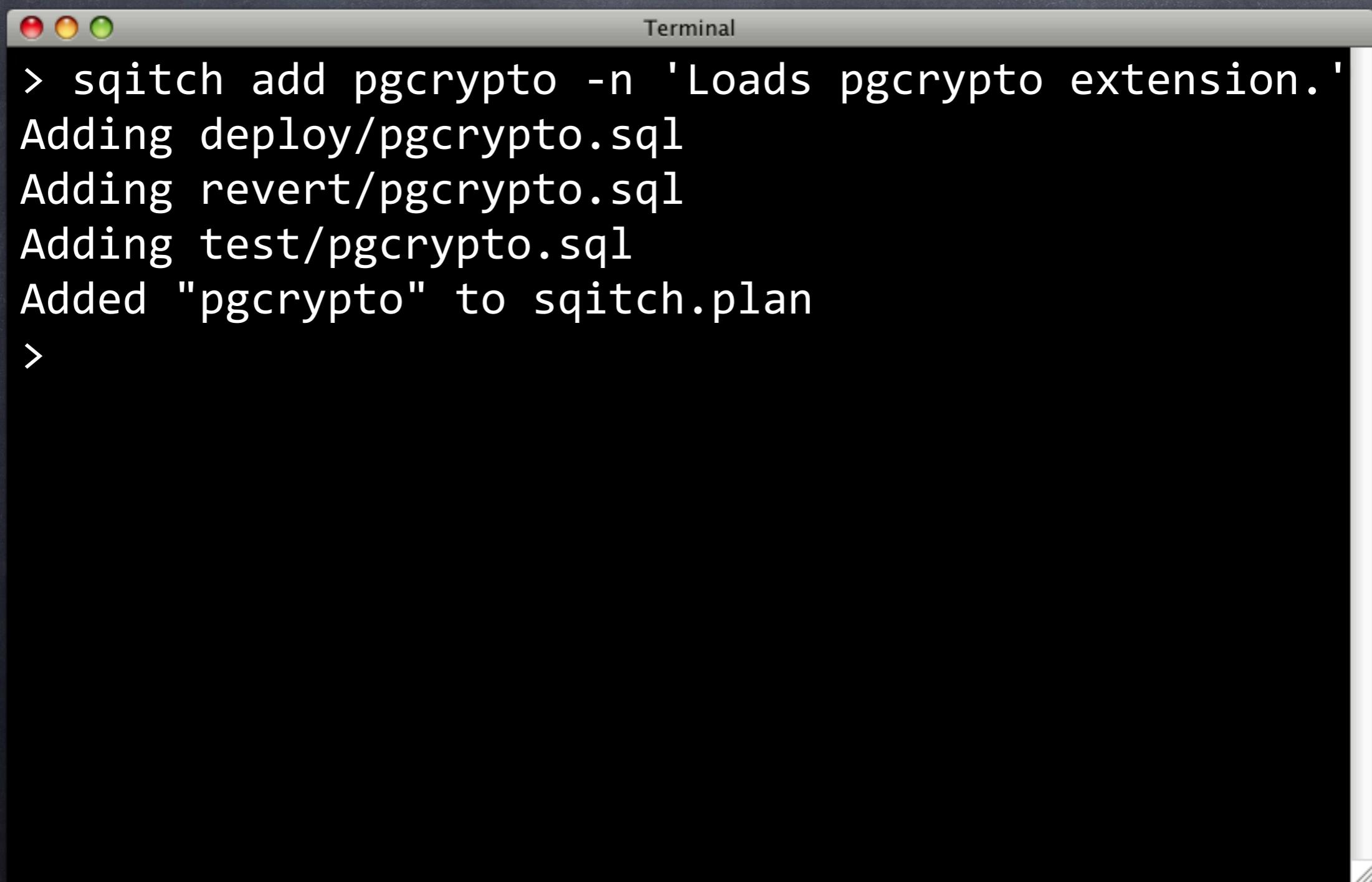
```
Terminal
> psql -d flipr_test -c "
    SELECT insert_user('foo', 'secr3t'),
           insert_user('bar', 'secr3t');
    SELECT nickname, password FROM users;
"
nickname | password
-----
foo      | 9695da4dd567a19f9b92065f240c6725
bar      | 9695da4dd567a19f9b92065f240c6725
```

Not good.

Add pgcrypto



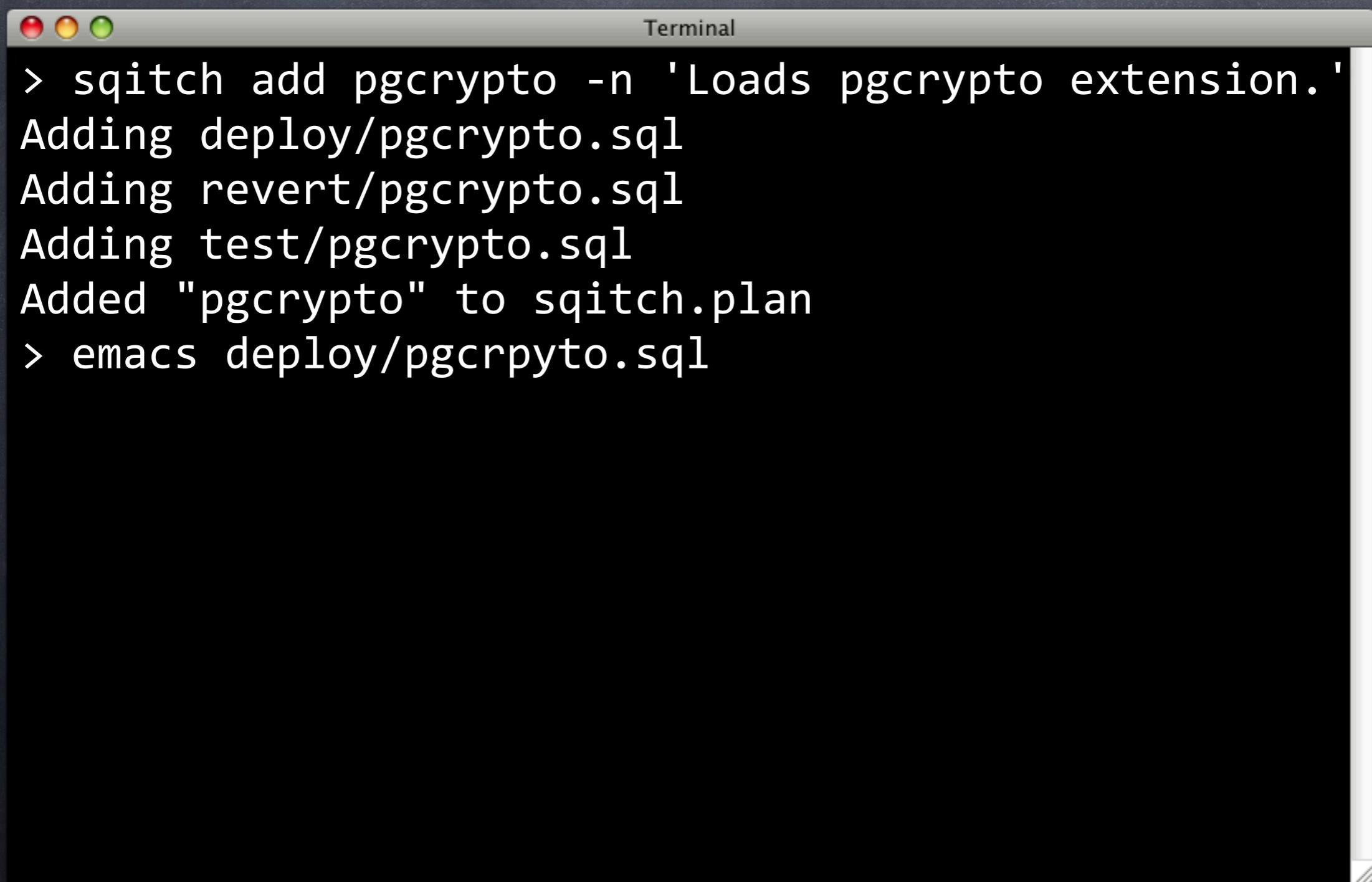
Add pgcrypto



A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The title bar reads "Terminal". The main pane contains the following text:

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

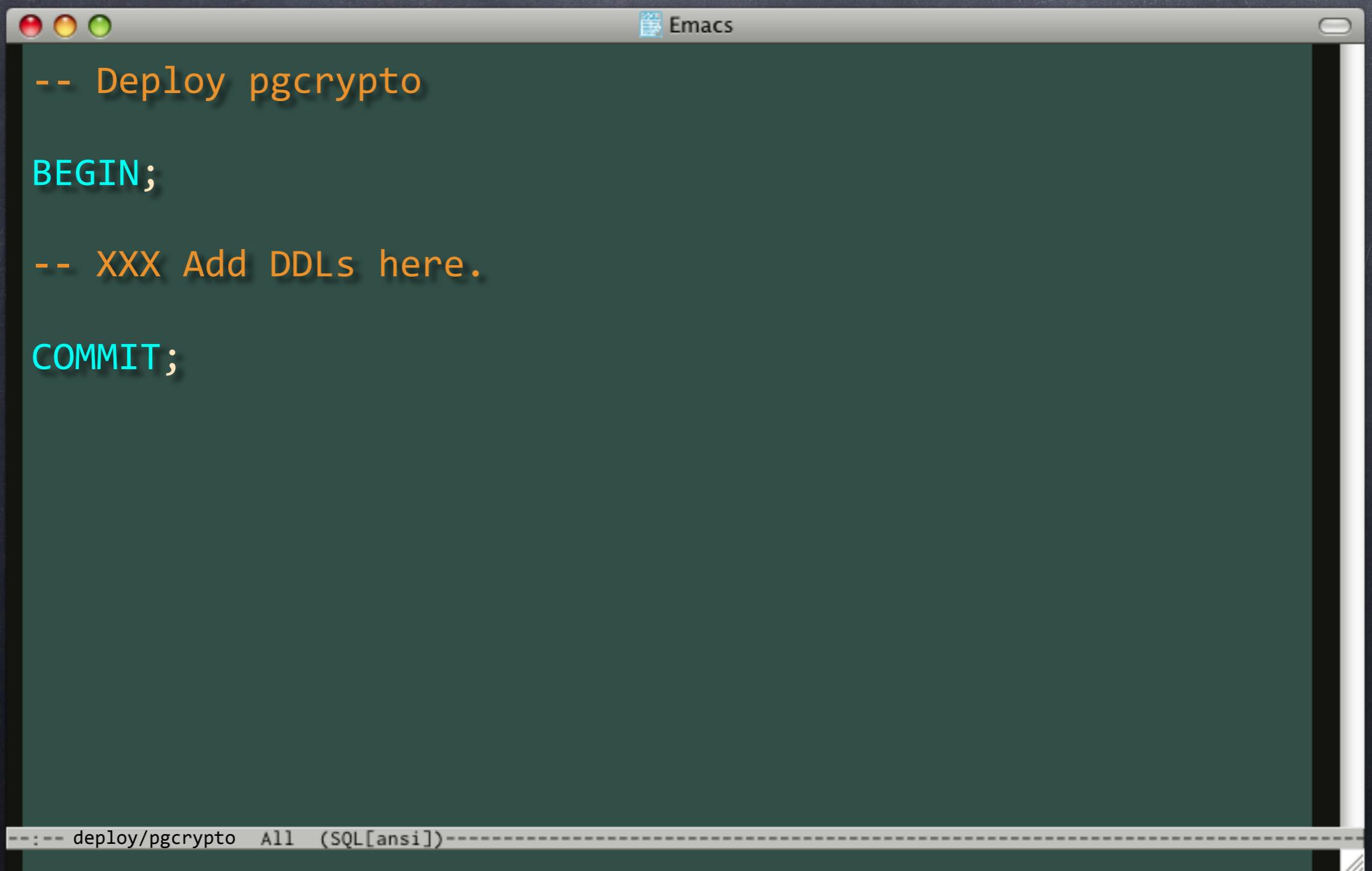
Add pgcrypto



A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The title bar reads "Terminal". The main pane contains the following text:

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

deploy/pgcrypto.sql



The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". The buffer contains the following SQL code:

```
-- Deploy pgcrypto

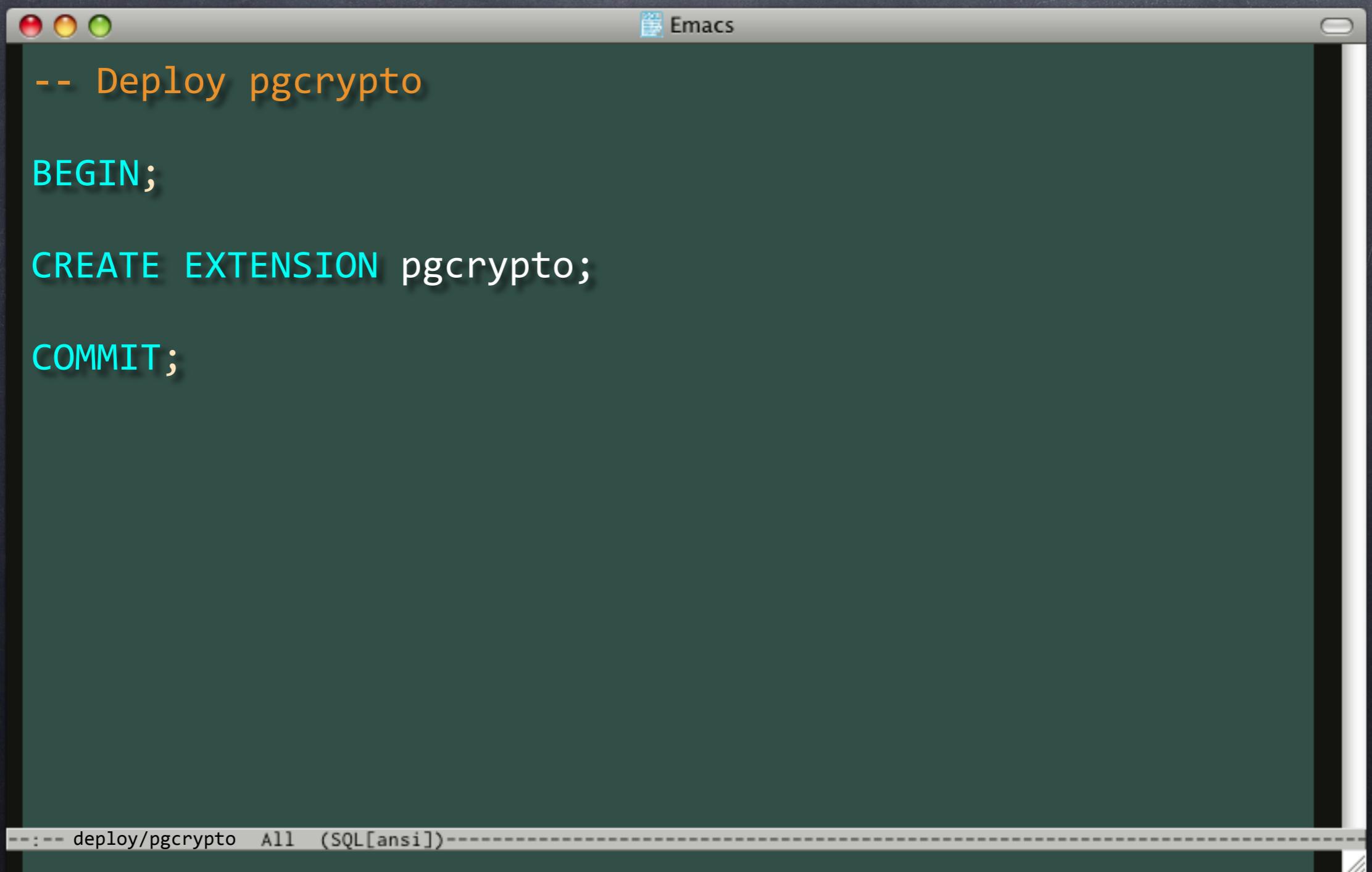
BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

At the bottom of the window, there is a status bar with the text "---- deploy/pgcrypto All (SQL[ansi])----".

deploy/pgcrypto.sql



The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". The buffer contains the following SQL code:

```
-- Deploy pgcrypto

BEGIN;

CREATE EXTENSION pgcrypto;

COMMIT;
```

At the bottom of the window, there is a status bar with the text "---- deploy/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 $$
+    INSERT INTO users values($1, md5($2));
+$$;
```

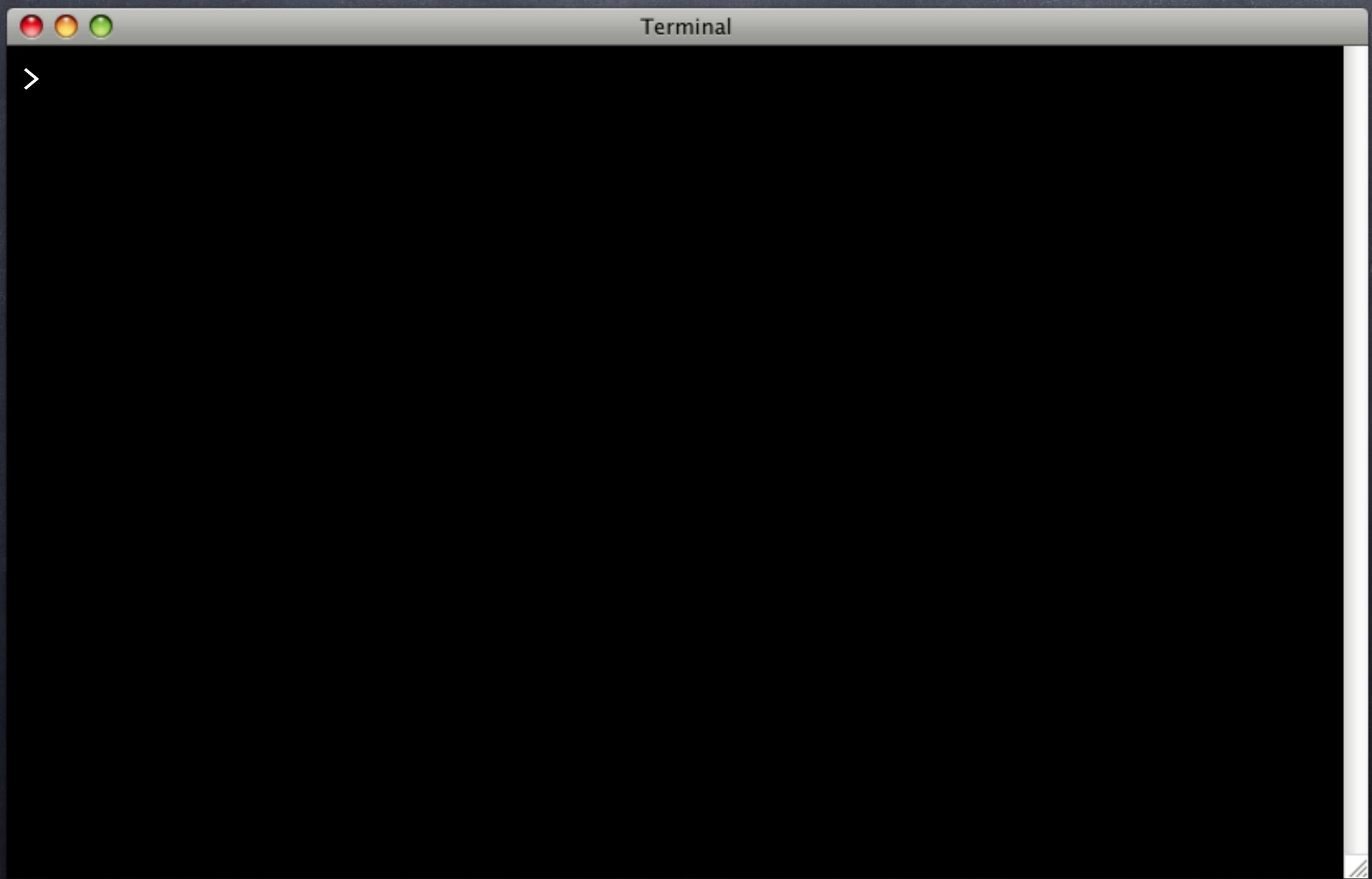


```
> git diff HEAD^
diff --git a/deploy/insert_user_crypt.sql b/deploy/insert_user_crypto.sql
new file mode 100644
index 0000000..fa8d0c6
--- /dev/null
+++ b/deploy/insert_user_crypt.sql
@@ -0,0 +1,8 @@
+-- requires: users, appuser, pgcrypto
+
+CREATE OR REPLACE FUNCTION insert_user(
+    nickname TEXT,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

0y.

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  
  * test/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  
  * test/insert_user.sql  
>
```

Rework It

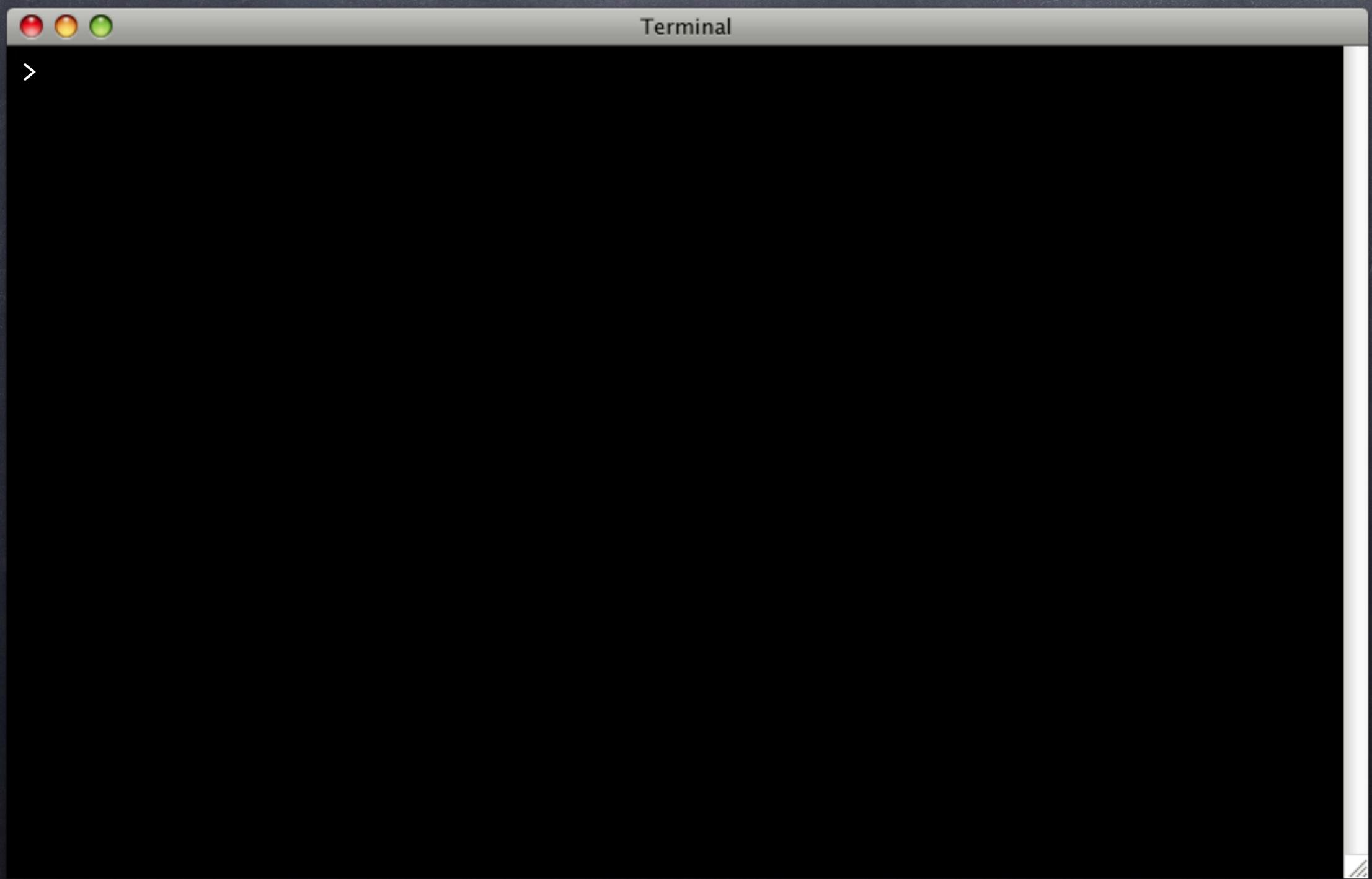
```
Terminal  
> sqitch rework insert_user -r pgcrypto \  
  -n 'Changes insert_user to use pgcrypto.'  
Added "insert_user [insert_user@v1.0.0-dev1 pgcrypto]" to  
sqitch.plan.  
Modify these files as appropriate:  
  * deploy/insert_user.sql  
  * revert/insert_user.sql  
  * test/insert_user.sql 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

Rework It

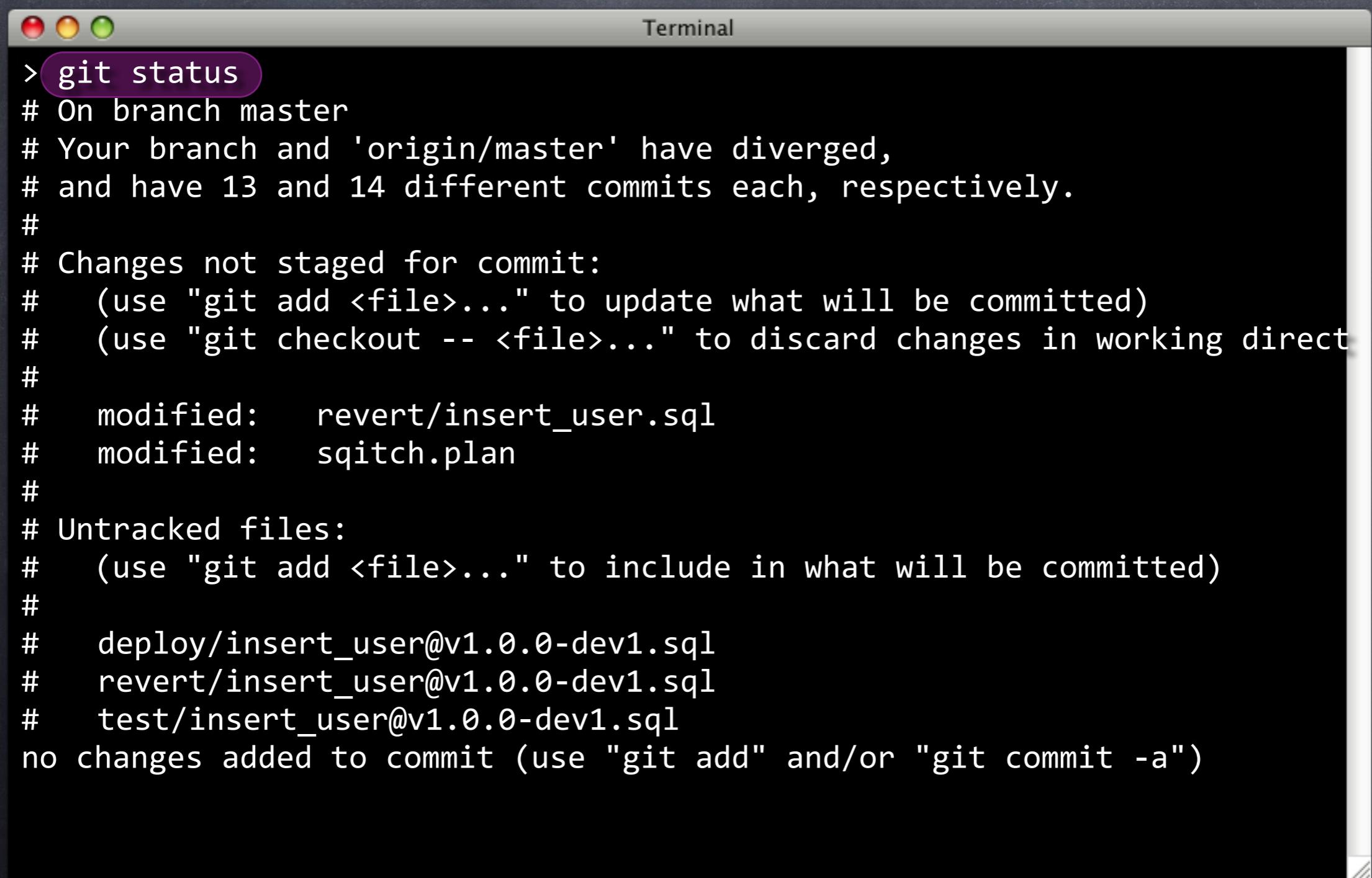
```
Terminal  
> sqitch rework insert_user -r pgcrypto \  
  -n 'Changes insert_user to use pgcrypto.'  
Added "insert_user [insert_user@v1.0.0-dev1 pgcrypto]" to  
sqitch.plan.  
Modify these files as appropriate:  
  * deploy/insert_user.sql  
  * revert/insert_user.sql  
  * test/insert_user.sql 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

Same files?

Same Files?



Same Files?



A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the output of a "git status" command. The output shows that the user is on the "master" branch and has diverged from the remote "origin/master" branch. It lists staged changes, unstaged changes (modified files: "revert/insert_user.sql" and "sqitch.plan"), and untracked files ("deploy/insert_user@v1.0.0-dev1.sql", "revert/insert_user@v1.0.0-dev1.sql", and "test/insert_user@v1.0.0-dev1.sql"). A purple oval highlights the command "git status".

```
> git status
# On branch master
# Your branch and 'origin/master' have diverged,
# and have 13 and 14 different commits each, respectively.
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working direct
#
#     modified:   revert/insert_user.sql
#     modified:   sqitch.plan
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#     deploy/insert_user@v1.0.0-dev1.sql
#     revert/insert_user@v1.0.0-dev1.sql
#     test/insert_user@v1.0.0-dev1.sql
no changes added to commit (use "git add" and/or "git commit -a")
```

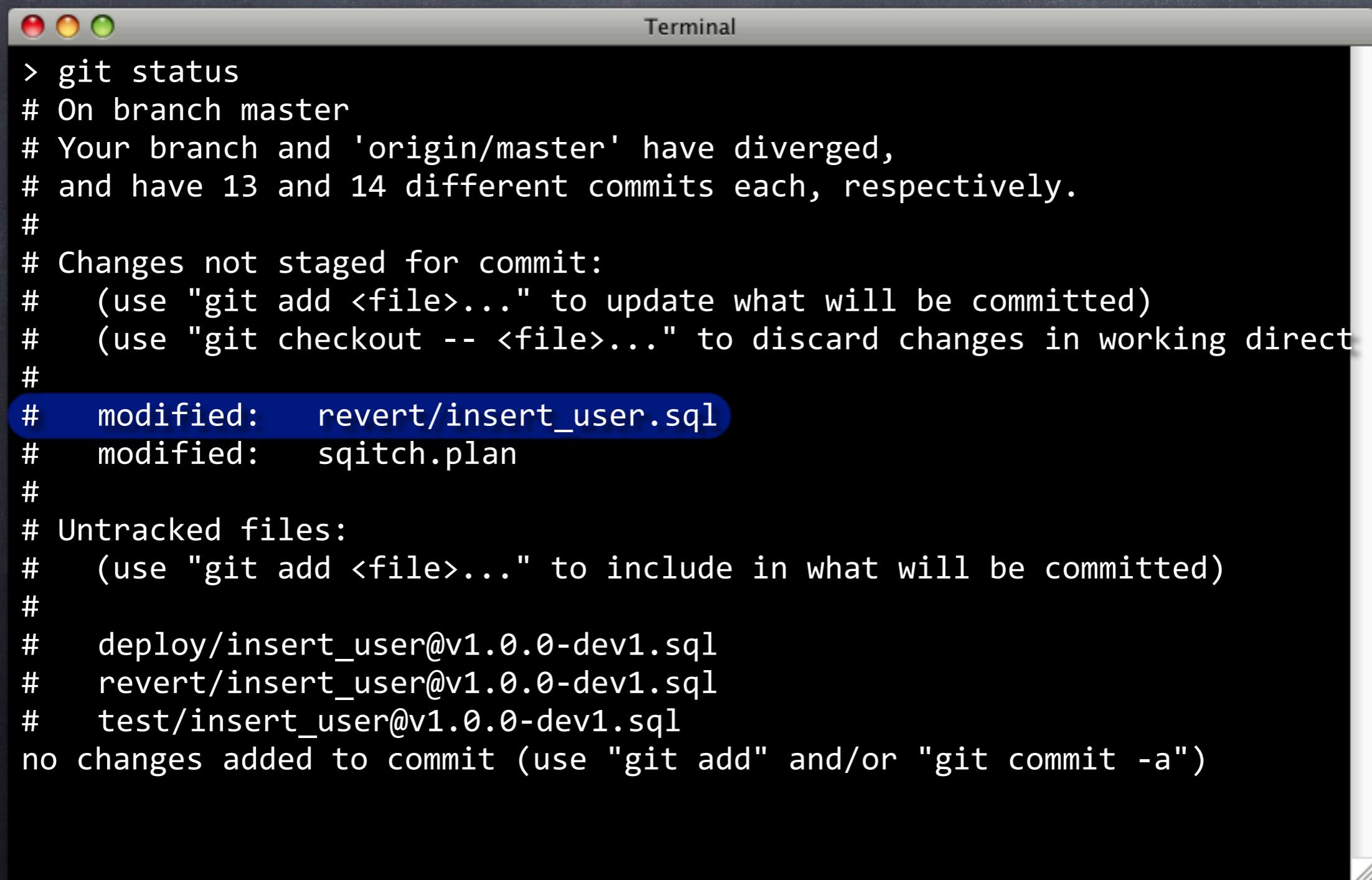
Same Files?

```
Terminal  
> git status  
# On branch master  
# Your branch and 'origin/master' have diverged,  
# and have 13 and 14 different commits each, respectively.  
#  
# Changes not staged for commit:  
#   (use "git add <file>..." to update what will be committed)  
#     (use "git checkout -- <file>..." to discard changes in working direct  
#  
#       modified:   revert/insert_user.sql  
#       modified:   sqitch.plan  
#  
# Untracked files:  
#   (use "git add <file>..." to include in what will be committed)  
#  
#   deploy/insert_user@v1.0.0-dev1.sql  
#   revert/insert_user@v1.0.0-dev1.sql  
#   test/insert_user@v1.0.0-dev1.sql  
no changes added to commit (use "git add" and/or "git commit -a")
```

Same Files?

```
Terminal  
> git status  
# On branch master  
# Your branch and 'origin/master' have diverged,  
# and have 13 and 14 different commits each, respectively.  
#  
# Changes not staged for commit:  
#   (use "git add <file>..." to update what will be committed)  
#     (use "git checkout -- <file>..." to discard changes in working direct  
#  
#       modified:   revert/insert_user.sql  
#       modified:   sqitch.plan  
#  
# Untracked files:  
#   (use "git add <file>..." to include in what will be committed)  
#  
#   deploy/insert_user@v1.0.0-dev1.sql  
#   revert/insert_user@v1.0.0-dev1.sql  
#   test/insert_user@v1.0.0-dev1.sql  
no changes added to commit (use "git add" and/or  
  
As of  
@v1.0.0-dev1
```

Same Files?



A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the output of a "git status" command. The output shows that the local branch "master" is behind the remote branch "origin/master" by one commit. It lists two modified files: "revert/insert_user.sql" and "sqitch.plan". There are also three untracked files: "deploy/insert_user@v1.0.0-dev1.sql", "revert/insert_user@v1.0.0-dev1.sql", and "test/insert_user@v1.0.0-dev1.sql". The text "no changes added to commit (use "git add" and/or "git commit -a")" is at the bottom.

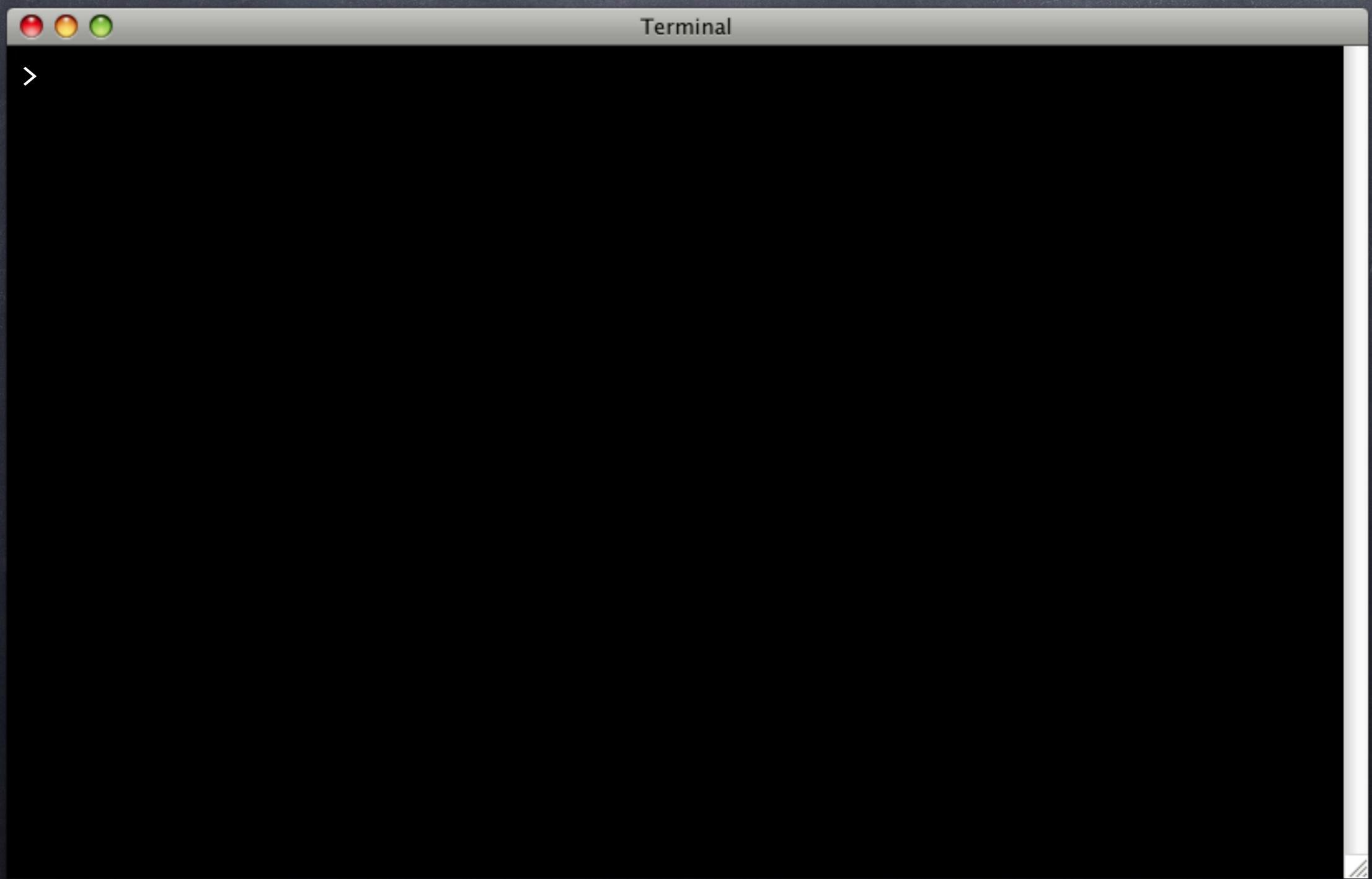
```
> git status
# On branch master
# Your branch and 'origin/master' have diverged,
# and have 13 and 14 different commits each, respectively.
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working direct
#
#       modified:   revert/insert_user.sql
#       modified:   sqitch.plan
#
# Untracked files:
#   (use "git add <file>..." to include in what will be committed)
#
#       deploy/insert_user@v1.0.0-dev1.sql
#       revert/insert_user@v1.0.0-dev1.sql
#       test/insert_user@v1.0.0-dev1.sql
no changes added to commit (use "git add" and/or "git commit -a")
```

Same Files?

```
Terminal  
> git status  
# On branch master  
# Your branch and 'origin/master' have diverged,  
# and have 13 and 14 different commits each, respectively.  
#  
# Changes not staged for commit:  
#   (use "git add <file>..." to update what will be committed)  
#   (use "git checkout -- <file>..." to discard changes in working direct  
#  
#       modified:   revert/insert_user.sql  
#       modified:   sqitch.plan  
#  
# Untracked files:  
#   (use "git add <file>..." to include in what will be committed)  
#  
#       deploy/insert_user@v1.0.0-dev1.sql  
#       revert/insert_user@v1.0.0-dev1.sql  
#       test/insert_user@v1.0.0-dev1.sql  
no changes added to commit (use "git add" and/or "git commit -a")
```

Previous
deploy becomes
revert

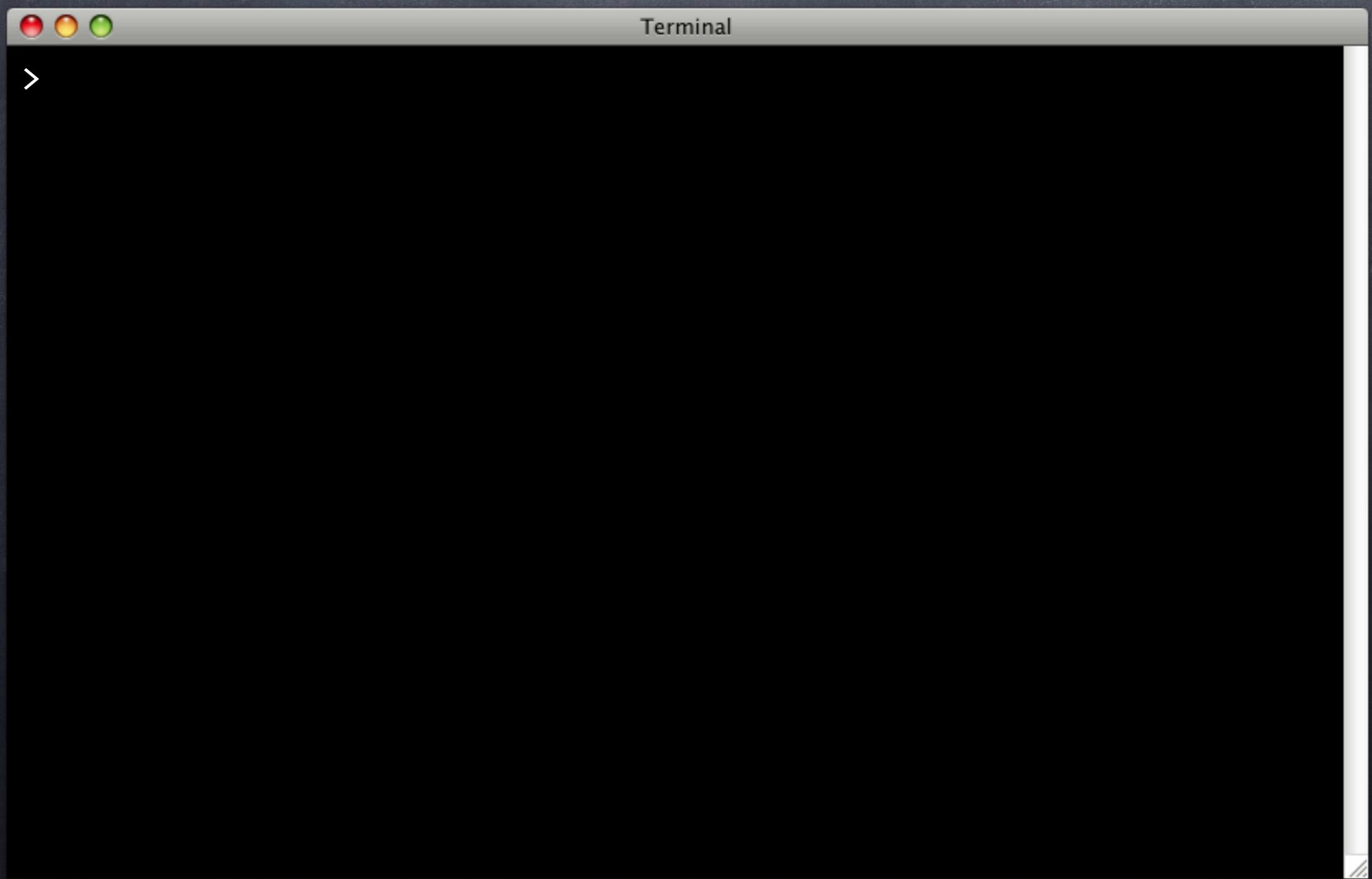
What's the Diff?



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  
* test/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  
  * test/change_pass.sql  
>
```

Rework change_pass

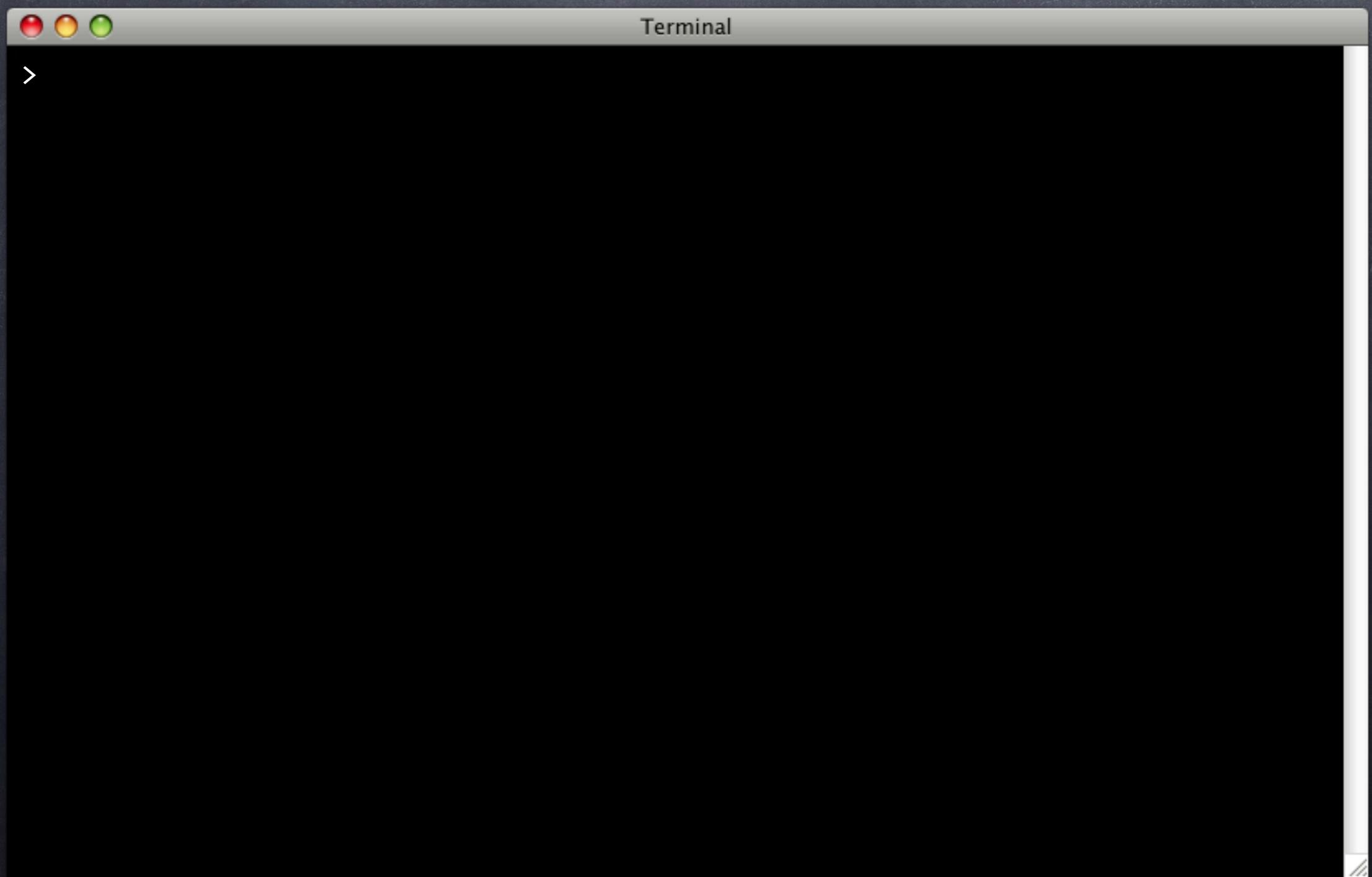
```
Terminal  
> sqitch rework change_pass -r pgcrypto \  
  -n 'Change change_pass to use pgcrypto.'  
Added "change_pass [change_pass@v1.0.0-dev1 pgcrypto]" to  
sqitch.plan.  
Modify these files as appropriate:  
  * deploy/change_pass.sql  
  * revert/change_pass.sql  
  * test/change_pass.sql  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

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  
  * test/change_pass.sql  
>
```

Same files?

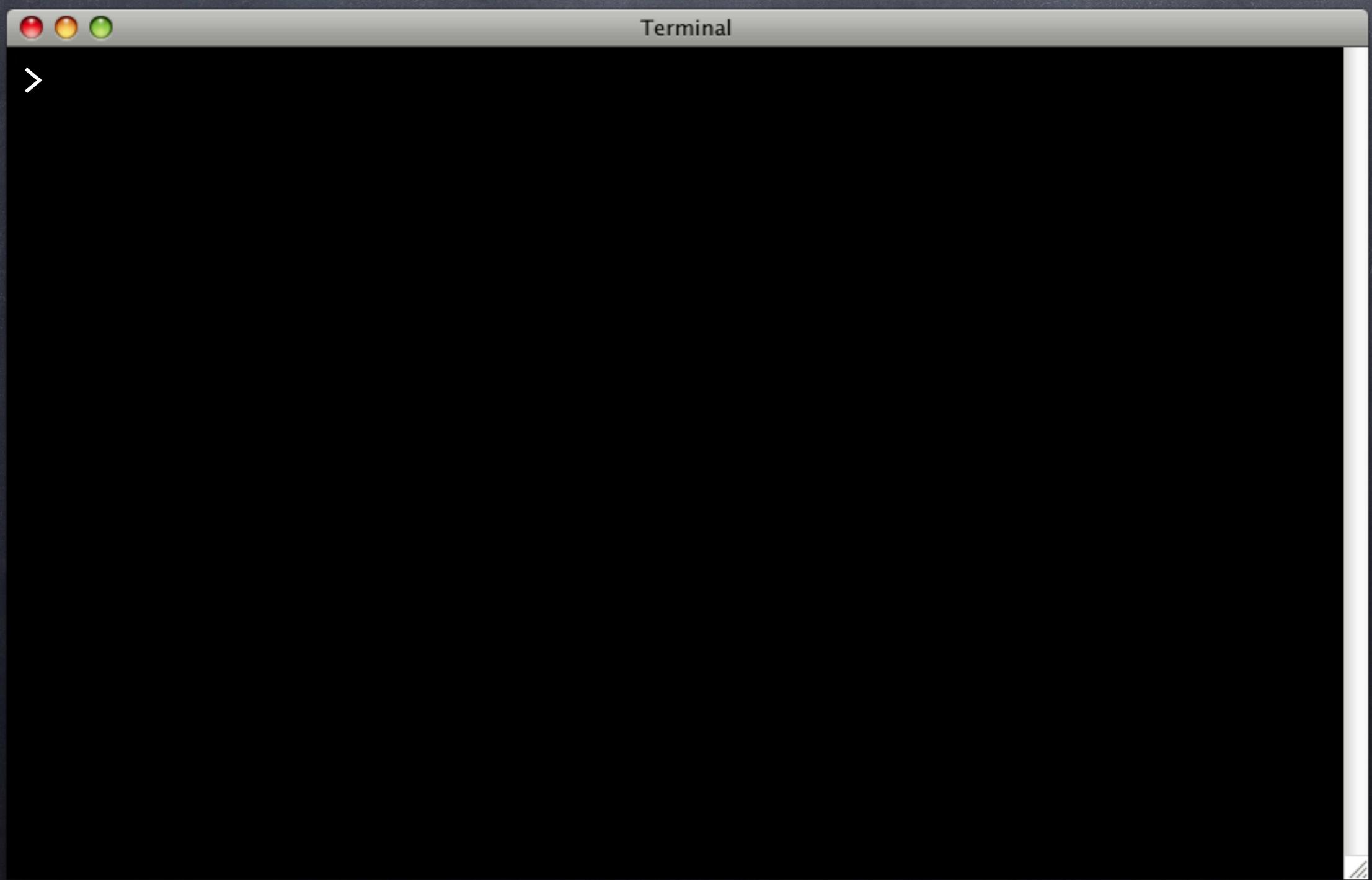
What's the Diff?



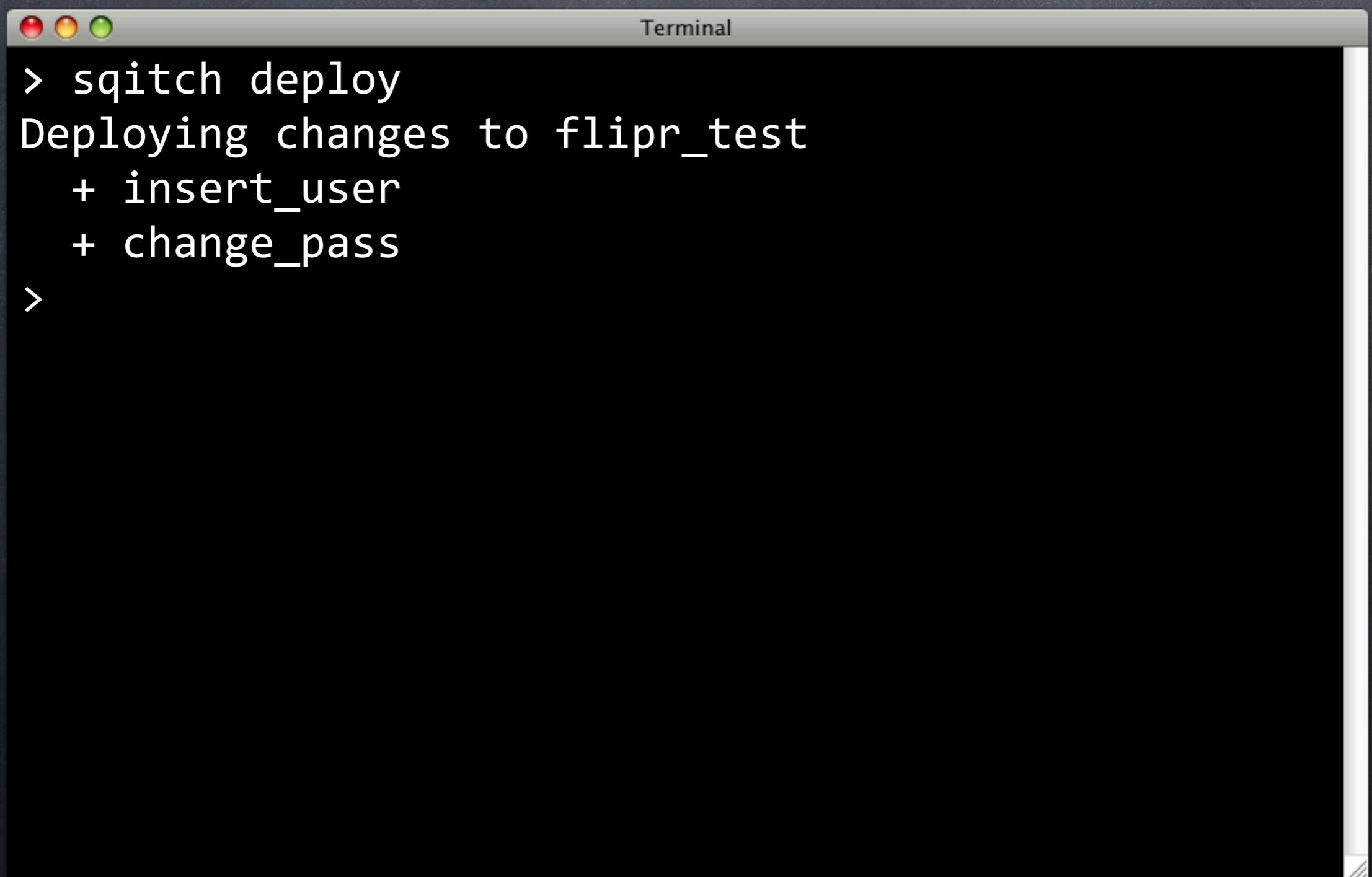
What's the Diff?

```
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!



A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The title bar reads "Terminal". The main pane contains the following text:

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

Send it Up!

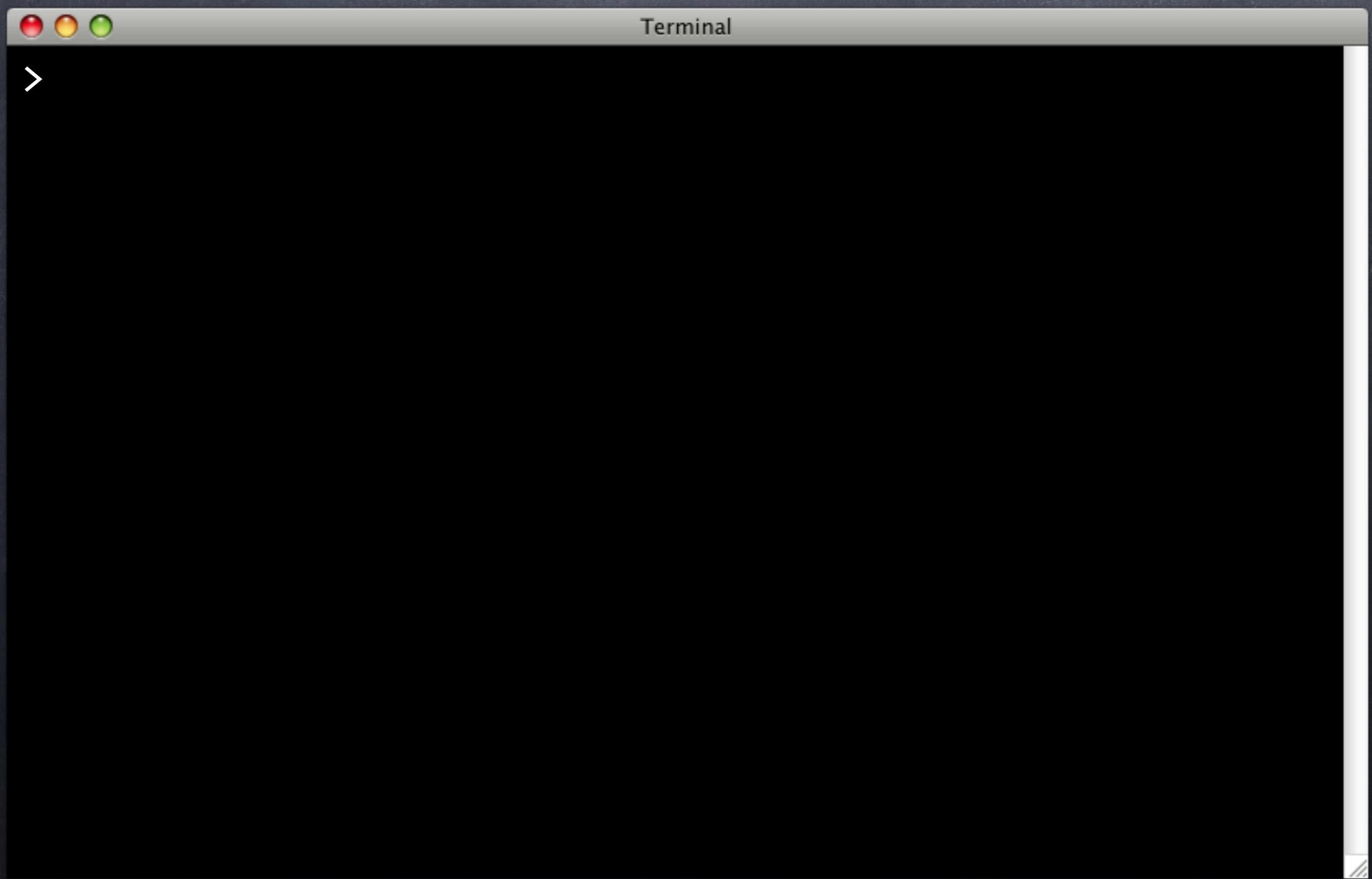
```
Terminal  
> sqitch deploy  
Deploying changes to flipr_test  
  + insert_user  
  + change_pass  
> psql -d flipr_test -c "  
    DELETE FROM users;  
    SELECT insert_user('foo', 'secr3t'),  
          insert_user('bar', 'secr3t');  
    SELECT nickname, password FROM users;  
"  
  
nickname | password  
-----+-----  
foo     | $1$160EKyF3$kv5ae7505ROub75d9QKTh/  
bar     | $1$J4NJDgaJ$578i9Lt6b8ohJwi6WhNN01  
>
```

Send it Up!

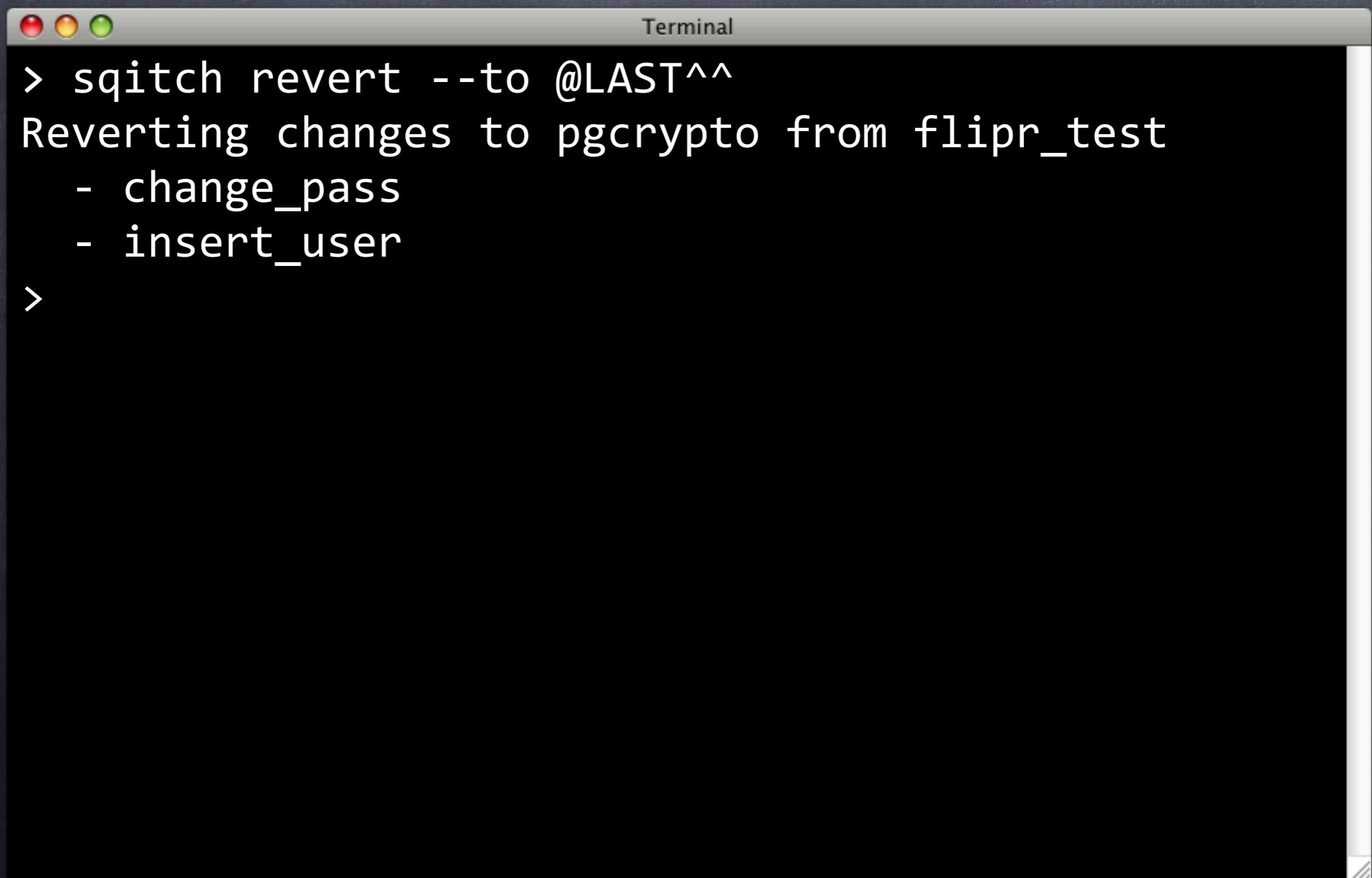
```
Terminal  
> sqitch deploy  
Deploying changes to flipr_test  
  + insert_user  
  + change_pass  
> psql -d flipr_test -c "  
    DELETE FROM users;  
    SELECT insert_user('foo', 'secr3t'),  
          insert_user('bar', 'secr3t');  
    SELECT nickname, password FROM users;  
"  
  
nickname | password  
-----+-----  
foo     | $1$160EKyF3$kv5ae7505ROub75d9QKTh/  
bar     | $1$J4NJDgaJ$578i9Lt6b8ohJwi6WhNN01  
>
```



Can We Go Back?



Can We Go Back?



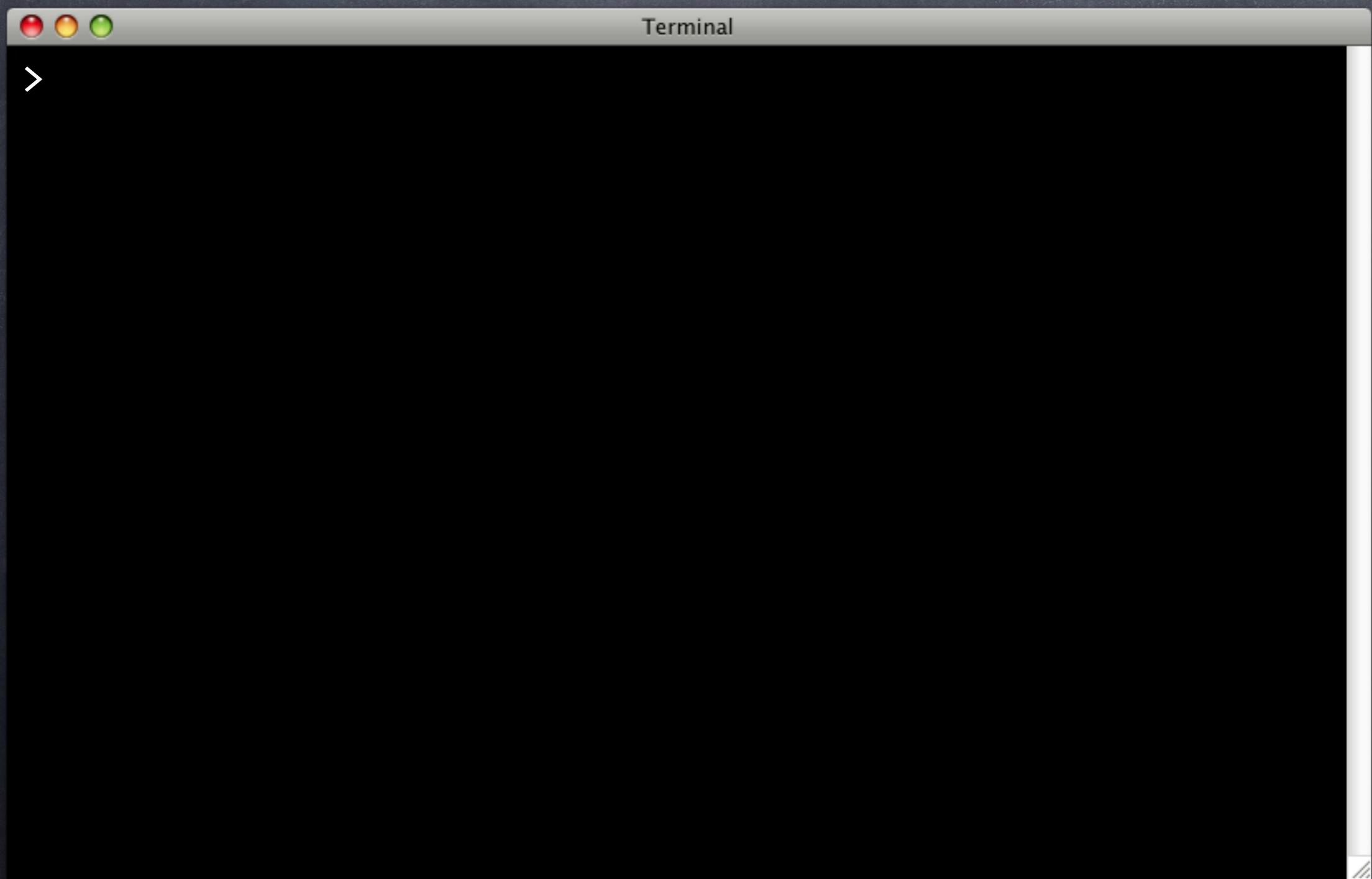
A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The terminal output is displayed in white text on a black background. The command entered was "sqitch revert --to @LAST^^". The output shows the process of reverting changes from a database named "pgcrypto" in a schema named "flipr_test". Two specific changes are listed as being reverted: "change_pass" and "insert_user". A final prompt ">" is shown at the bottom.

```
> sqitch revert --to @LAST^^
Reverting changes to pgcrypto from flipr_test
- change_pass
- insert_user
>
```

Can We Go Back?

```
Terminal
> sqitch revert --to @LAST^^
Reverting changes to pgcrypto from flipr_test
- change_pass
- insert_user
> 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
>
```

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  
>
```

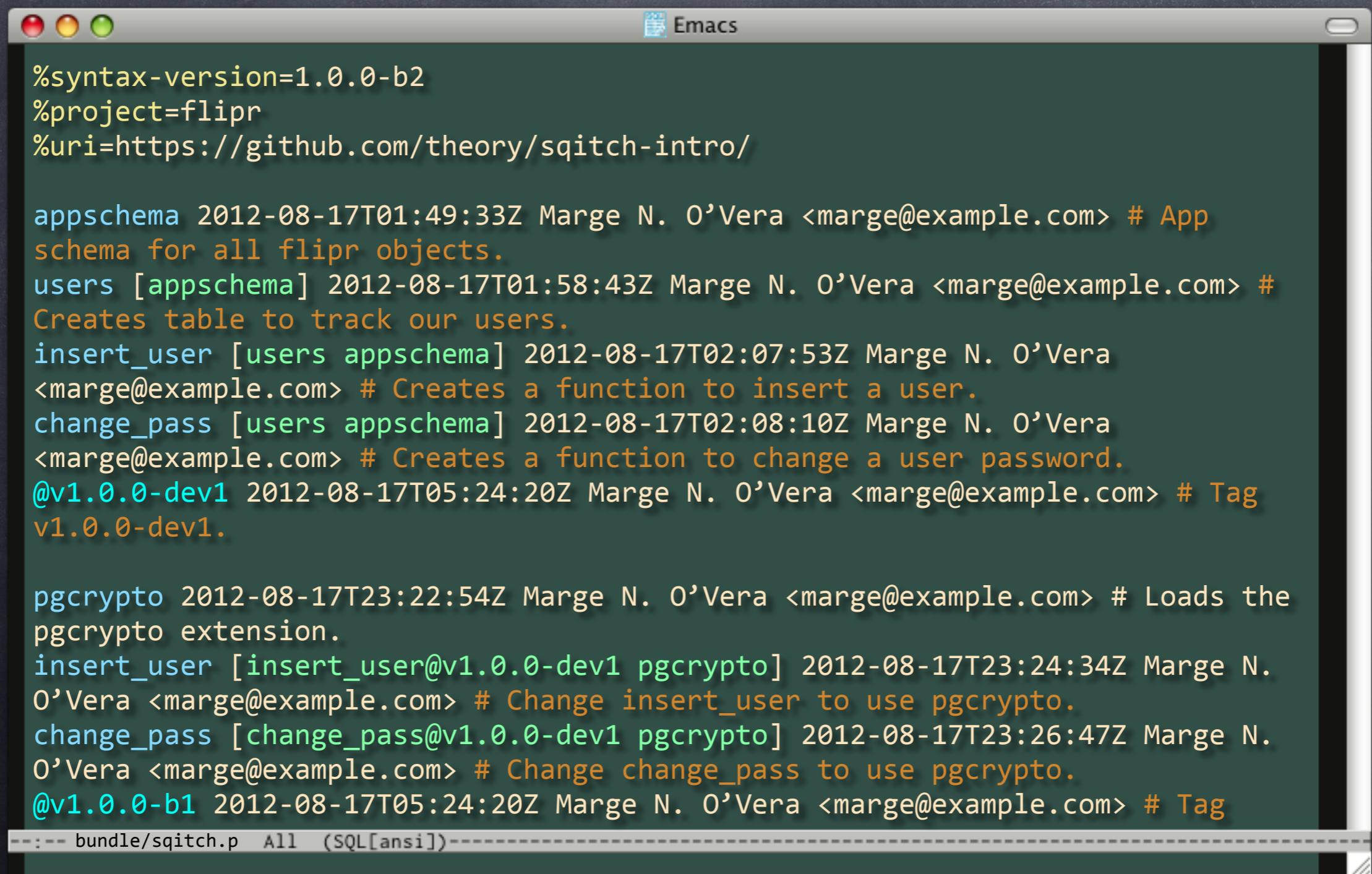
What About Bundling?

```
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?



The screenshot shows an Emacs window with a dark green background and white text. The title bar reads "Emacs". The window contains a sqitch history log. The log entries are color-coded: blue for schema names, green for table names, and orange for comments and function definitions. The log includes details like commit date, author, email, and commit message.

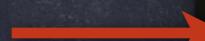
```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-08-17T01:49:33Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-08-17T01:58:43Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-08-17T02:07:53Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-08-17T02:08:10Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
@v1.0.0-dev1 2012-08-17T05:24:20Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2012-08-17T23:22:54Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2012-08-17T23:24:34Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2012-08-17T23:26:47Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2012-08-17T05:24:20Z Marge N. O'Vera <marge@example.com> # Tag

----- bundle/sqitch.p All (SQL[ansi]) -----
```

What's the Plan?



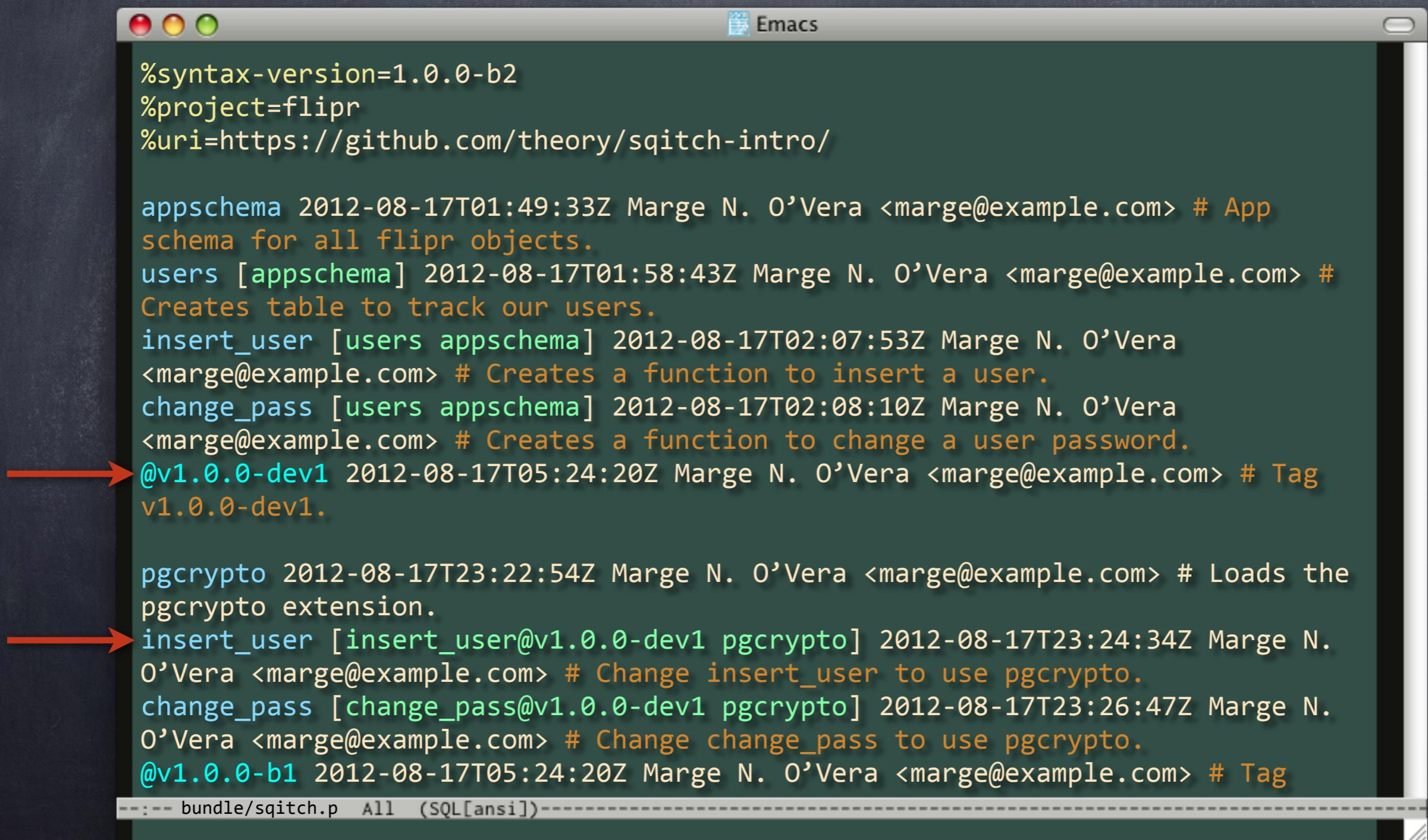
```
%syntax-version=1.0.0-b2
%project=flipr
%uri=https://github.com/theory/sqitch-intro/

appschema 2012-08-17T01:49:33Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-08-17T01:58:43Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-08-17T02:07:53Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-08-17T02:08:10Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
@v1.0.0-dev1 2012-08-17T05:24:20Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2012-08-17T23:22:54Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2012-08-17T23:24:34Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2012-08-17T23:26:47Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2012-08-17T05:24:20Z Marge N. O'Vera <marge@example.com> # Tag

---- bundle/sqitch.p All (SQL[ansi])----
```

What's the Plan?



The screenshot shows an Emacs window with a dark green background and white text. The title bar says "Emacs". The buffer contains a sqitch log file with the following content:

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

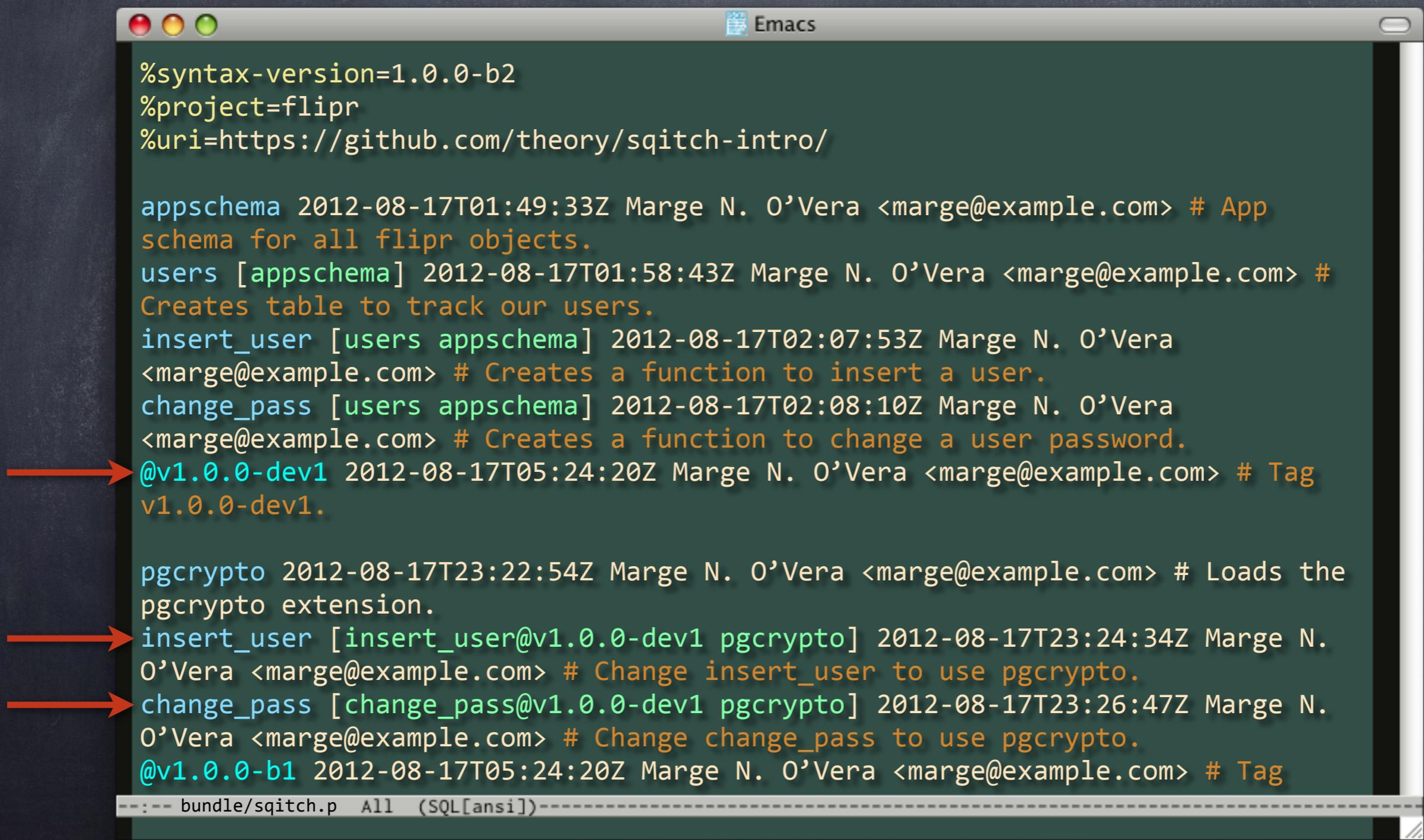
appschema 2012-08-17T01:49:33Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-08-17T01:58:43Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-08-17T02:07:53Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-08-17T02:08:10Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
→@v1.0.0-dev1 2012-08-17T05:24:20Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2012-08-17T23:22:54Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
→insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2012-08-17T23:24:34Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2012-08-17T23:26:47Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2012-08-17T05:24:20Z Marge N. O'Vera <marge@example.com> # Tag

---- bundle/sqitch.p All (SQL[ansi])----
```

Two red arrows point to the first two entries in the log, specifically to the tag markers "→@v1.0.0-dev1" and "→insert_user [insert_user@v1.0.0-dev1 pgcrypto]".

What's the Plan?



The screenshot shows an Emacs window with a dark green background and white text. The title bar says "Emacs". The buffer contains a sqitch log file with the following content:

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

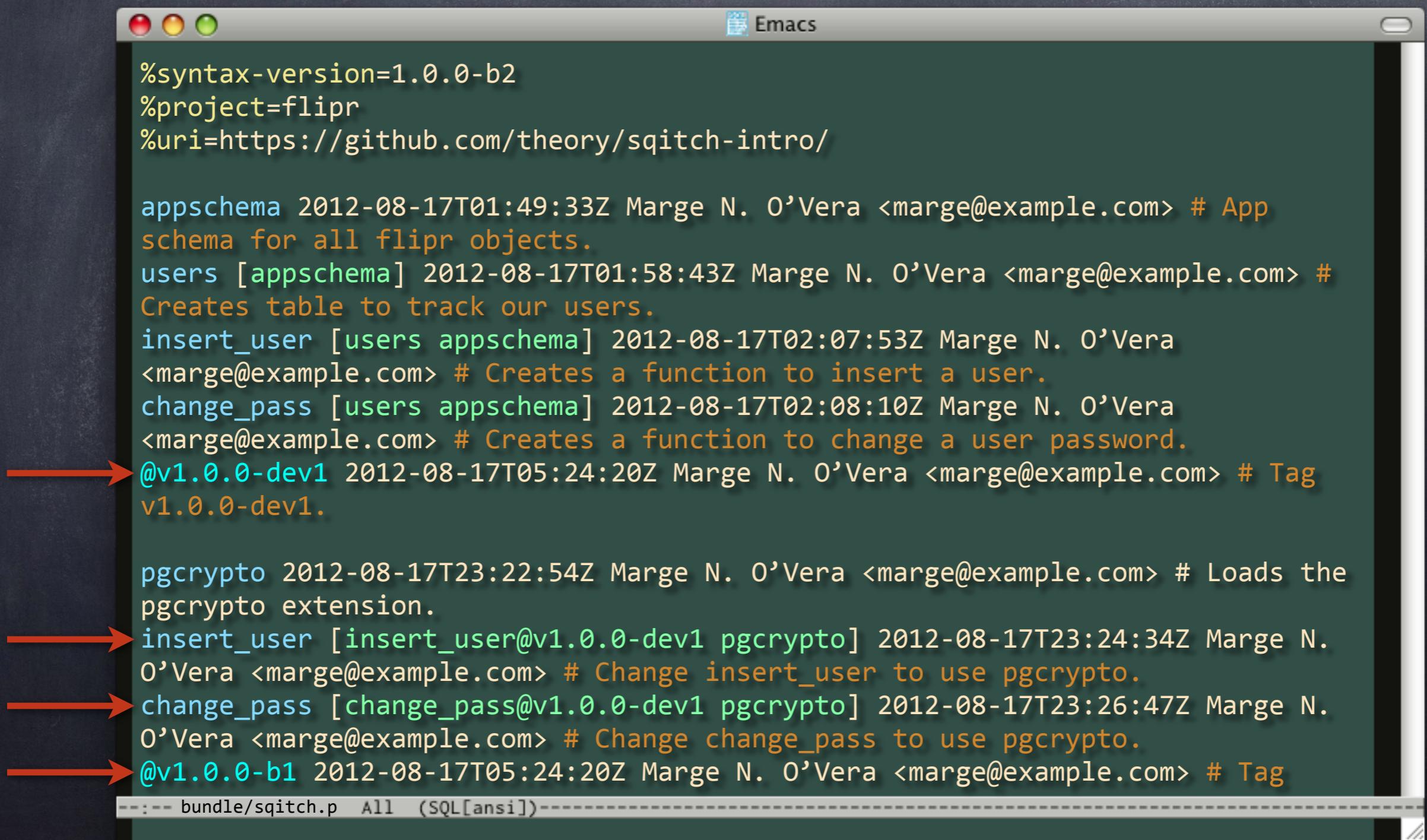
appschema 2012-08-17T01:49:33Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-08-17T01:58:43Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-08-17T02:07:53Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-08-17T02:08:10Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
→@v1.0.0-dev1 2012-08-17T05:24:20Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2012-08-17T23:22:54Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
→insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2012-08-17T23:24:34Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
→change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2012-08-17T23:26:47Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
@v1.0.0-b1 2012-08-17T05:24:20Z Marge N. O'Vera <marge@example.com> # Tag

----- bundle/sqitch.p All (SQL[ansi]) -----
```

Three red arrows point to the log entries for "insert_user", "change_pass", and "pgcrypto", indicating they are the focus of the discussion.

What's the Plan?



The screenshot shows an Emacs window with a dark green background and white text. The title bar says "Emacs". The buffer contains a sqitch log file with the following content:

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

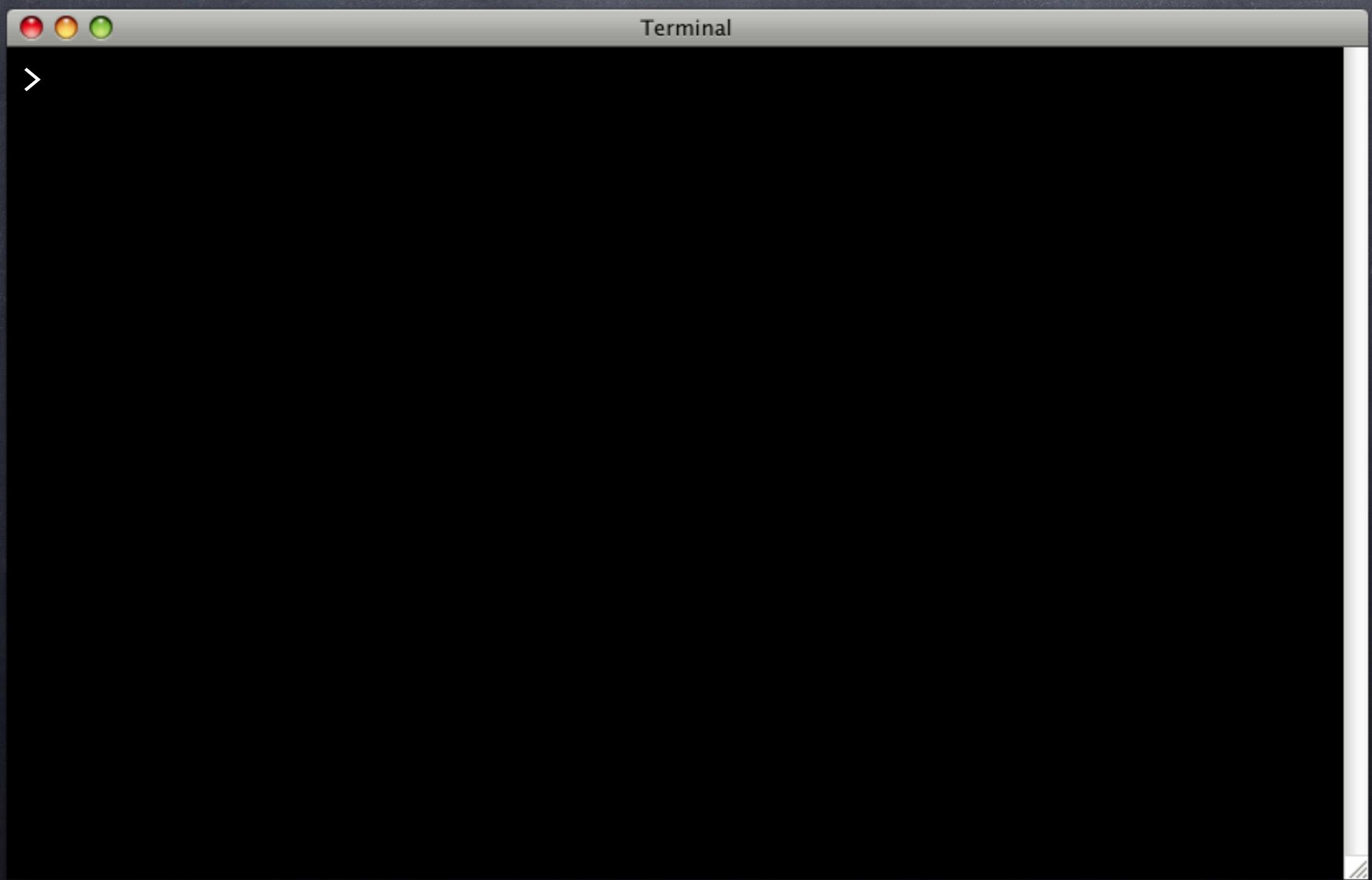
appschema 2012-08-17T01:49:33Z Marge N. O'Vera <marge@example.com> # App
schema for all flipr objects.
users [appschema] 2012-08-17T01:58:43Z Marge N. O'Vera <marge@example.com> #
Creates table to track our users.
insert_user [users appschema] 2012-08-17T02:07:53Z Marge N. O'Vera
<marge@example.com> # Creates a function to insert a user.
change_pass [users appschema] 2012-08-17T02:08:10Z Marge N. O'Vera
<marge@example.com> # Creates a function to change a user password.
→@v1.0.0-dev1 2012-08-17T05:24:20Z Marge N. O'Vera <marge@example.com> # Tag
v1.0.0-dev1.

pgcrypto 2012-08-17T23:22:54Z Marge N. O'Vera <marge@example.com> # Loads the
pgcrypto extension.
→insert_user [insert_user@v1.0.0-dev1 pgcrypto] 2012-08-17T23:24:34Z Marge N.
O'Vera <marge@example.com> # Change insert_user to use pgcrypto.
→change_pass [change_pass@v1.0.0-dev1 pgcrypto] 2012-08-17T23:26:47Z Marge N.
O'Vera <marge@example.com> # Change change_pass to use pgcrypto.
→@v1.0.0-b1 2012-08-17T05:24:20Z Marge N. O'Vera <marge@example.com> # Tag

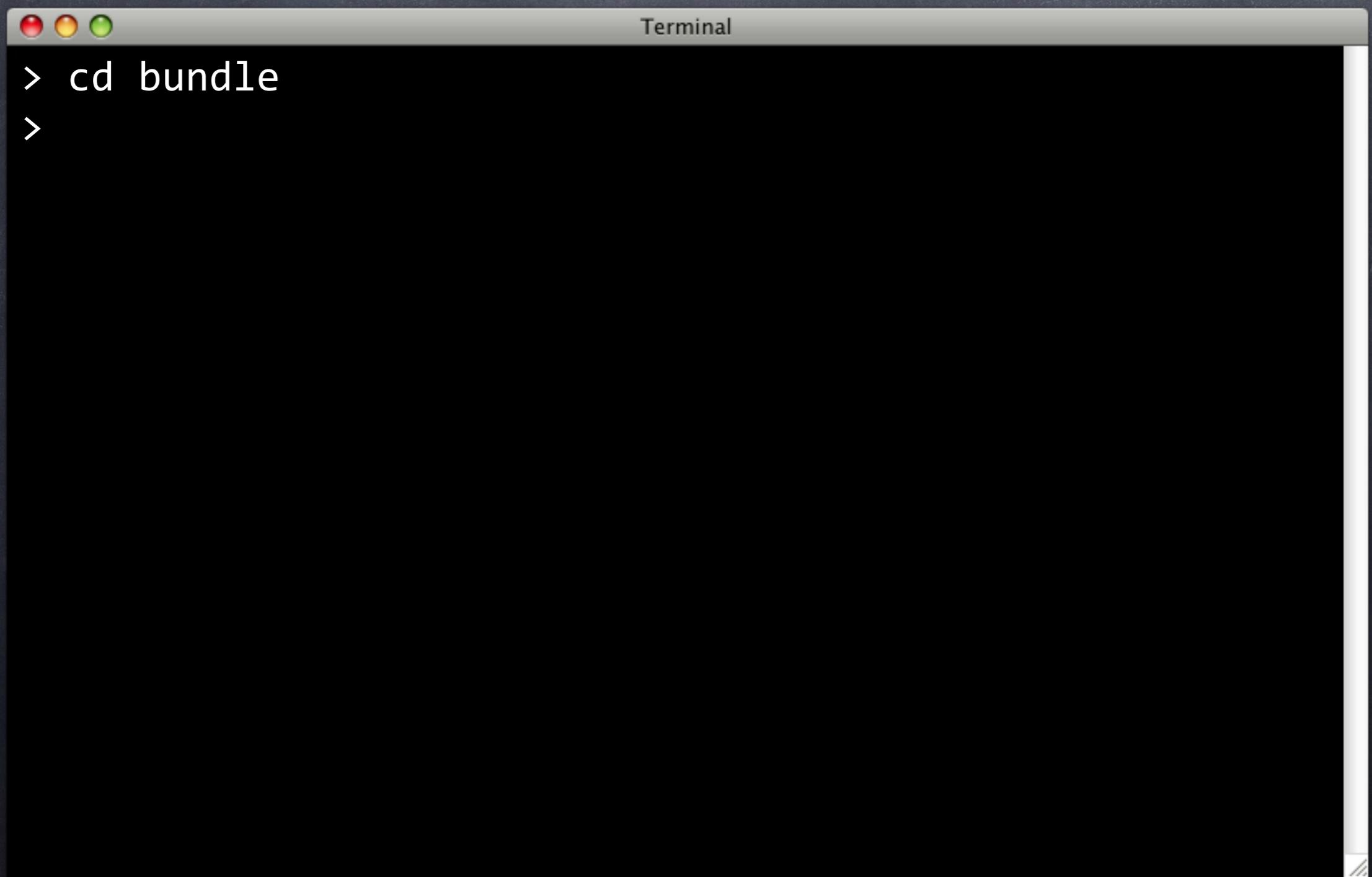
----- bundle/sqitch.p All (SQL[ansi]) -----
```

Four red arrows point to the log entries for the `insert_user`, `change_pass`, and `@v1.0.0-b1` tags, indicating they are the focus of the presentation.

Make it So

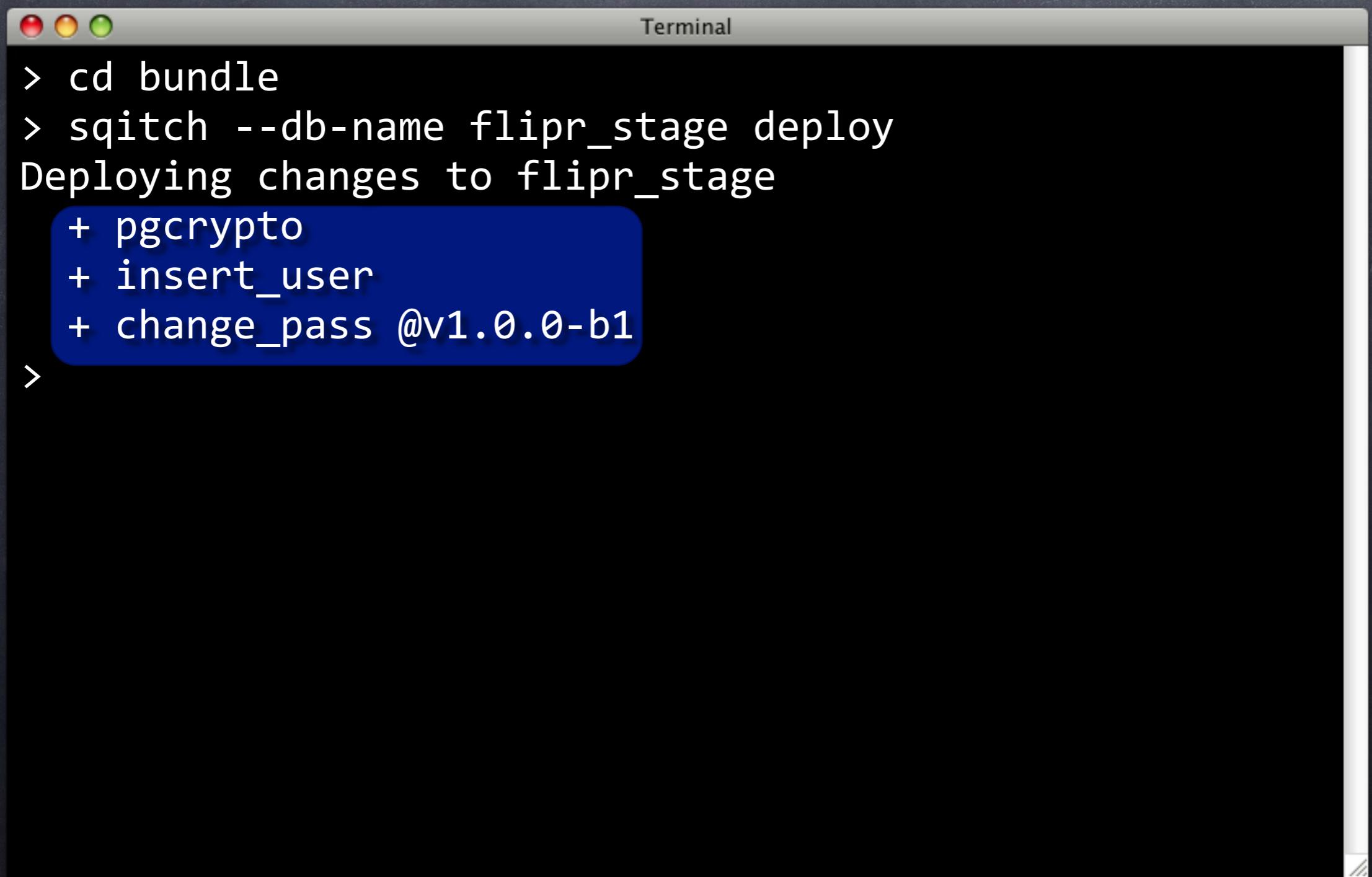


Make it So



```
Terminal  
> cd bundle  
>
```

Make it So



A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the following text:

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

The last three lines of the output are highlighted with a blue rounded rectangle.

Make it So

```
Terminal  
> cd bundle  
> sqitch --db-name flipr_stage deploy  
Deploying changes to flipr_stage  
+ pgcrypto  
+ insert_user  
+ change_pass @v1.0.0-b1  
> sqitch --db-name flipr_stage status  
# On database flipr_stage  
# Project: flipr  
# Change: d0eb9c5ff822877696e834d686a4d9ee8a5cd2a9  
# Name: change_pass  
# Tag: @v1.0.0-b1  
# Deployed: 2012-08-17 13:42:09 -0700  
# 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  
+ insert_user  
+ change_pass @v1.0.0-b1  
> sqitch --db-name flipr_stage status  
# On database flipr_stage  
# Project: flipr  
# Change: d0eb9c5ff822877696e834d686a4d9ee8a5cd2a9  
# Name: change_pass  
# Tag: @v1.0.0-b1  
# Deployed: 2012-08-17 13:42:09 -0700  
# 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
- ⦿ check - Validate plan & database

Other Commands

- help - Get it
- log - Like git log
- check - Validate plan & database
- test - Test deployment success



Current Status



Current Status

- ➊ Adding locale support



Current Status

- ➊ Adding locale support
- ➋ Porting to SQLite



Current Status

- ⦿ Adding locale support
- ⦿ Porting to SQLite
- ⦿ Adding VCS/SCM integration



Current Status

- ⦿ Adding locale support
- ⦿ Porting to SQLite
- ⦿ Adding VCS/SCM integration
- ⦿ Sync tags



Current Status

- ⦿ Adding locale support
- ⦿ Porting to SQLite
- ⦿ Adding VCS/SCM integration
 - ⦿ Sync tags
 - ⦿ Fetch reworked files from history



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/>

PDXPUG
September 20, 2012



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.

