

Simple SQL Change Management with Sqitch

David E. Wheeler

PGCon 2012
Ottawa, Canada



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

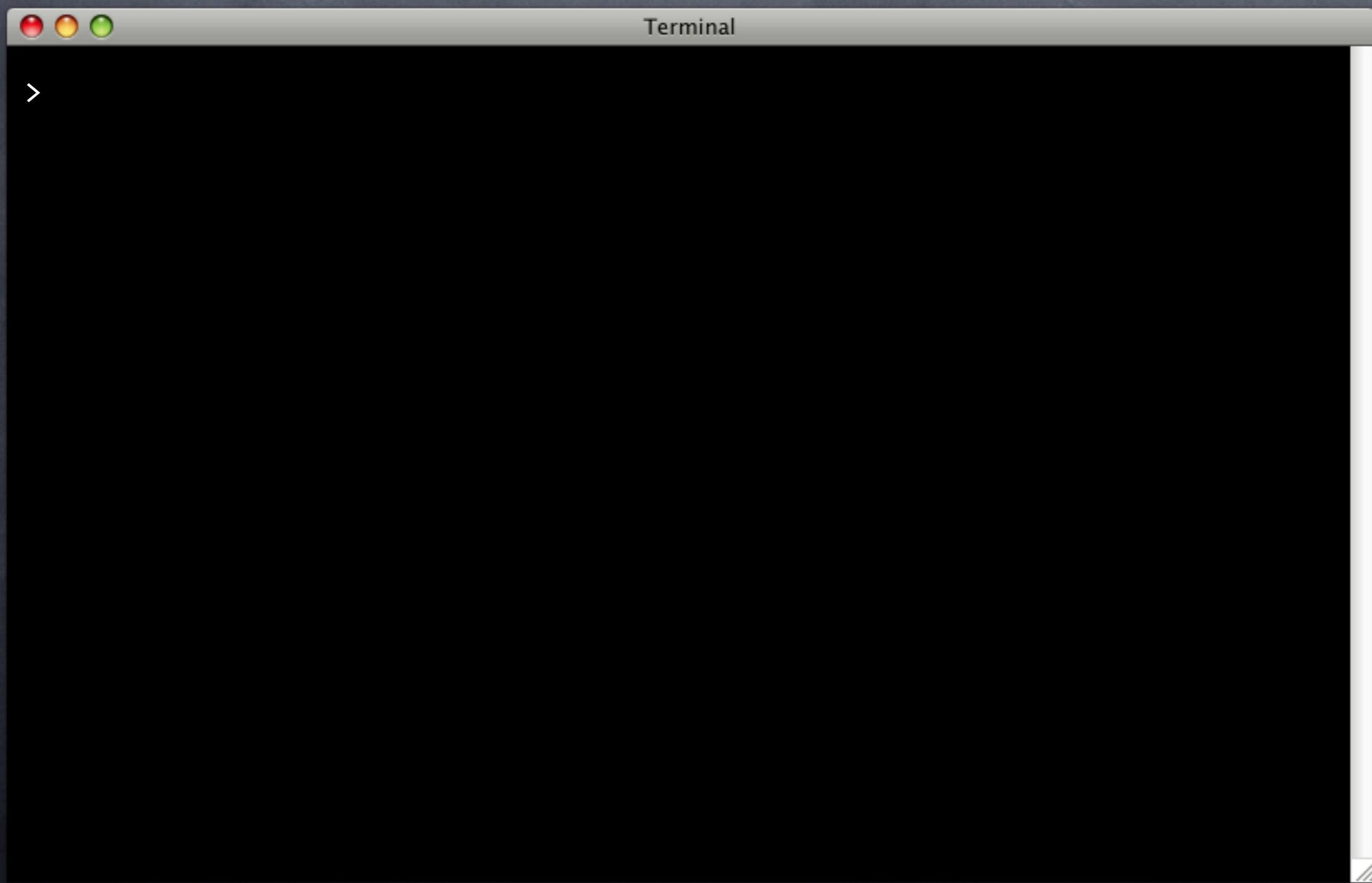
What about SQL Migrations?

- ~~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
- Managing procedures is a PITA

Imagine this Change



Imagine this Change

```
Terminal

> git diff
diff --git a/sql/deploy/recur.sql b/sql/deploy/recur.sql
index 622d52e..56e419e 100644
--- a/sql/deploy/recur.sql
+++ b/sql/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/sql/deploy/recur.sql b/sql/deploy/recur.sql  
index 622d52e..56e419e 100644  
--- a/sql/deploy/recur.sql  
+++ b/sql/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

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

Not So Much

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

Not So Much

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

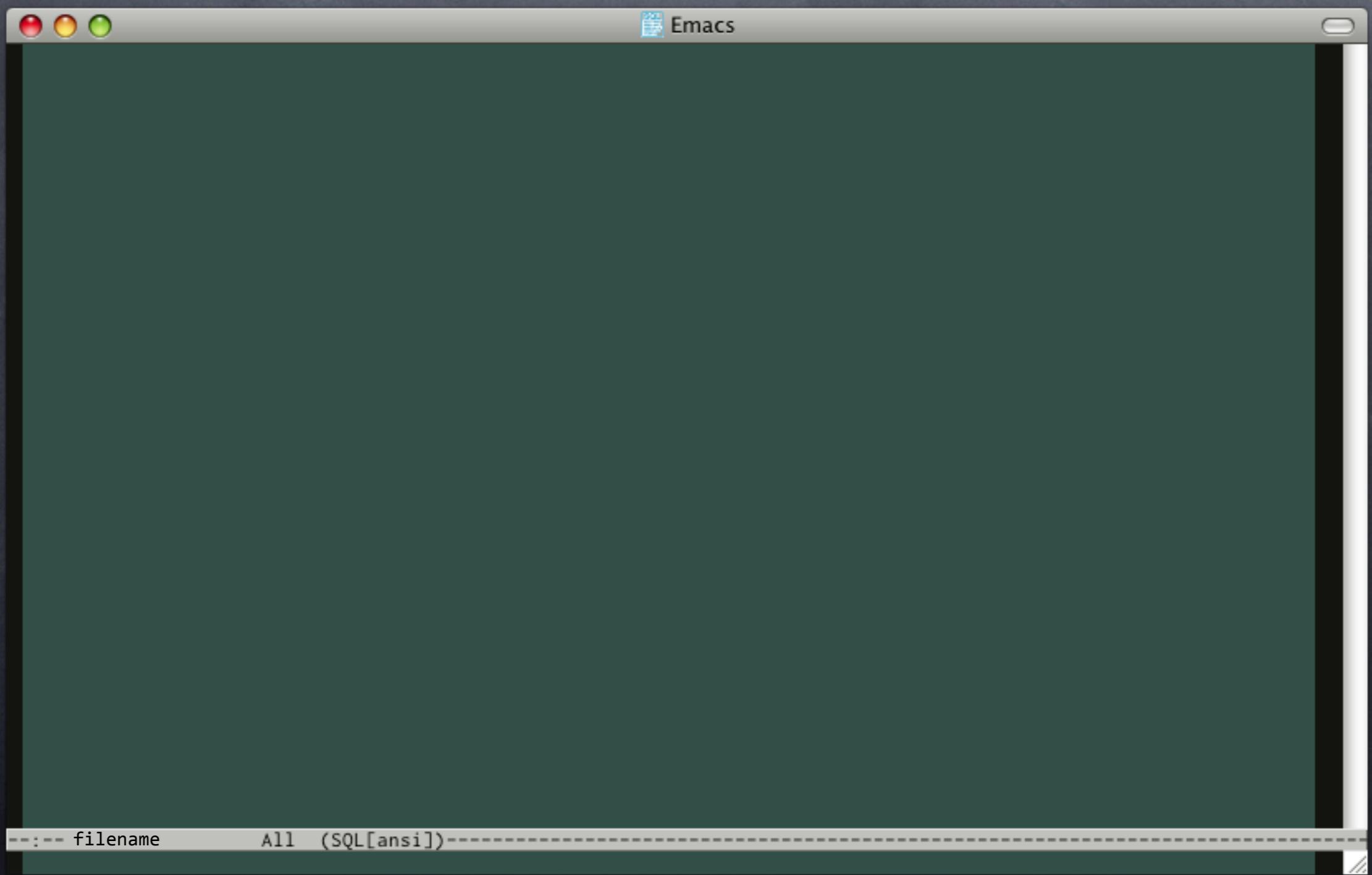
Not So Much

- 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

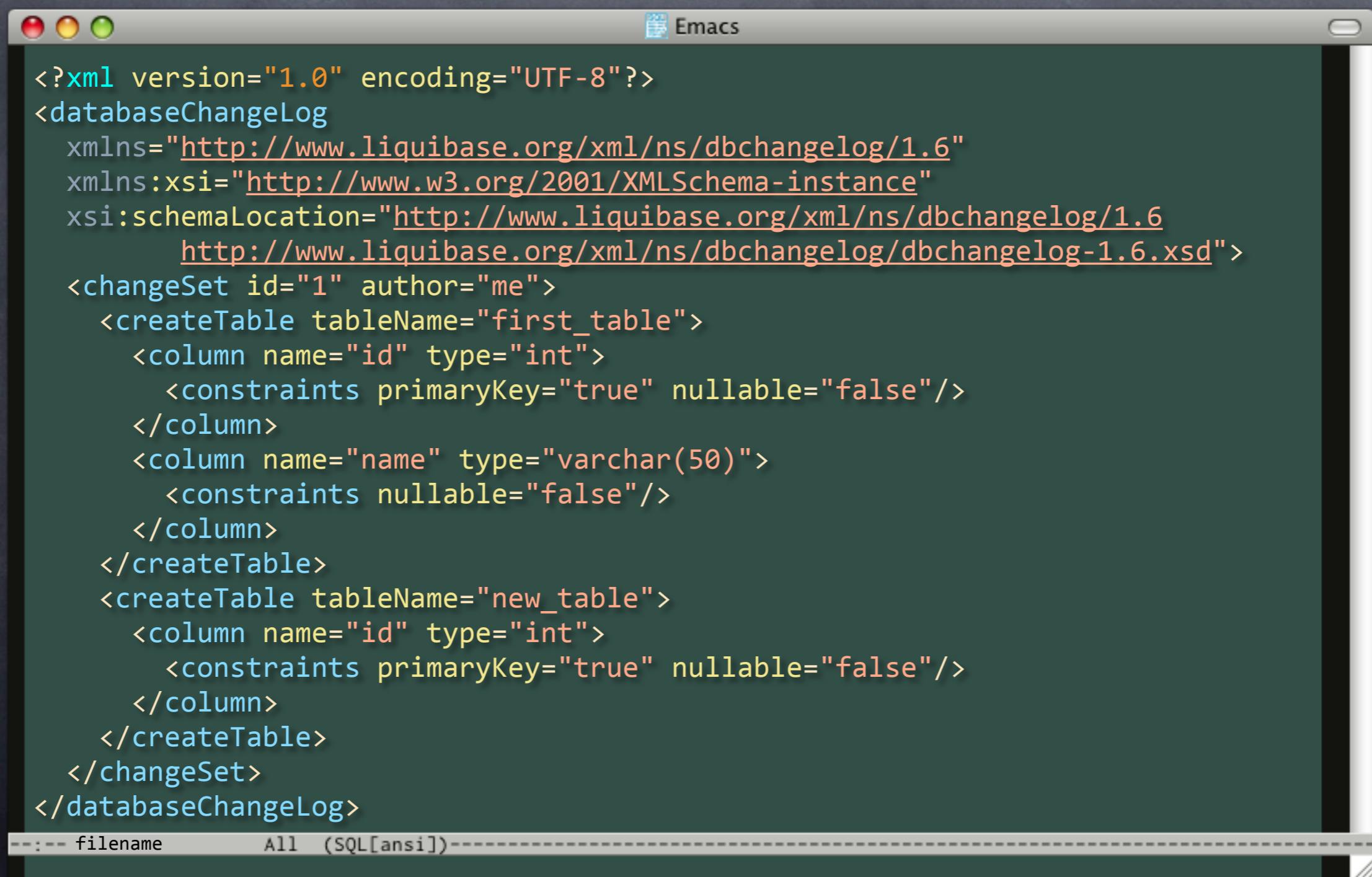
Not So Much

- ⦿ 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, with tags in blue and attributes in red. The buffer status at the bottom indicates it's a SQL file.

```
<?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

- ⦿ No opinions
- ⦿ Native scripting

Sqitch Philosophy

- No opinions
- Native scripting
- VCS integration

Sqitch Philosophy

- No opinions
- Native scripting
- VCS integration
- Dependency resolution

Sqitch Philosophy

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

Sqitch Philosophy

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

Sqitch Philosophy

Sqitch Philosophy

- Reduced duplication

Sqitch Philosophy

- Reduced duplication
- 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

- step

Sqitch Terminology

- step
- tag

Sqitch Terminology

- step
- tag
- plan

Sqitch Terminology

- step
- tag
- plan
- deploy

Sqitch Terminology

- step
- tag
- plan
- deploy
- revert

Caveats

Caveats



Caveats

- ➊ Under heavy development



Caveats

- Under heavy development
- v0.30 Testing release today



Caveats

- Under heavy development
- v0.30 Testing release today
- Functionality rapidly evolving



Caveats

- ⦿ Under heavy development
- ⦿ v0.30 Testing release today
- ⦿ Functionality rapidly evolving
- ⦿ Some gaps still to be filled in

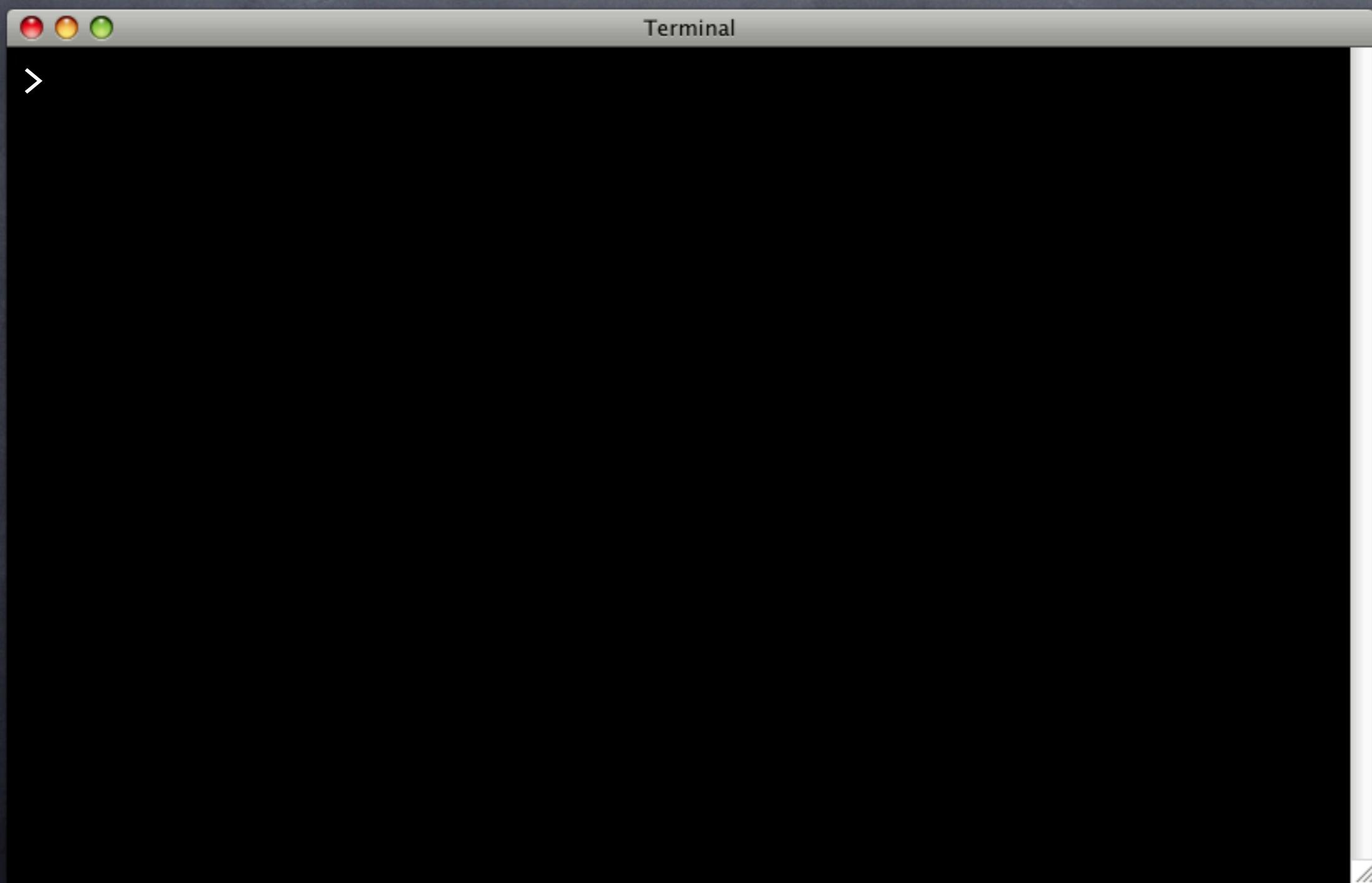


Caveats

- ⦿ Under heavy development
- ⦿ v0.30 Testing release today
- ⦿ Functionality rapidly evolving
- ⦿ Some gaps still to be filled in
- ⦿ VCS integration in flux



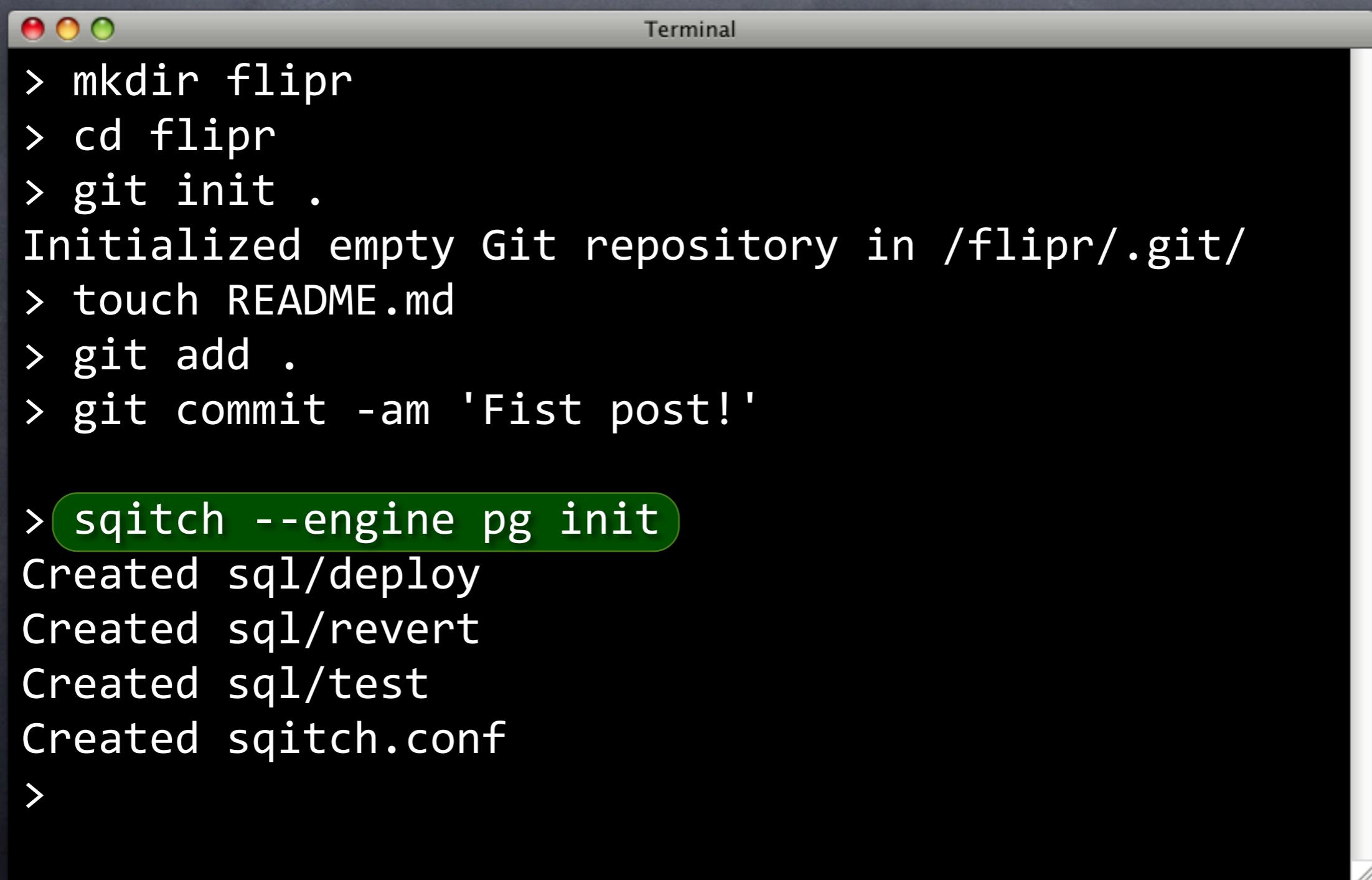
How it Works



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



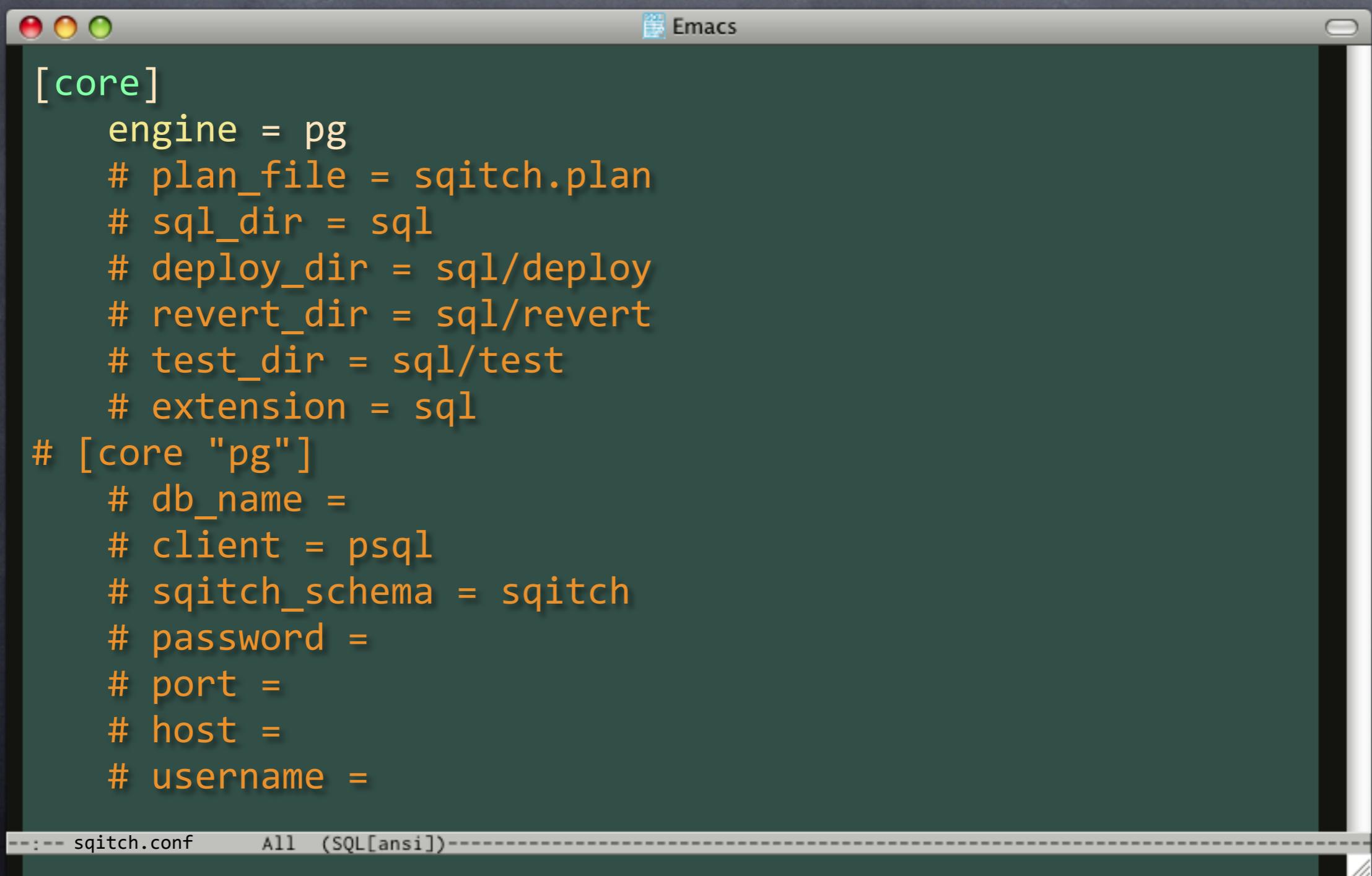
```
Terminal
> mkdir flipr
> cd flipr
> git init .
Initialized empty Git repository in /flipr/.git/
> touch README.md
> git add .
> git commit -am 'First post!'

> sqitch --engine pg init
Created sql/deploy
Created sql/revert
Created sql/test
Created sqitch.conf
>
```

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  
Created sql/deploy  
Created sql/revert  
Created sql/test  
Created sqitch.conf  
> emacs sqitch.conf
```

sqitch.conf



The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "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 a [core] section and a commented-out [core "pg"] section.

```
[core]
  engine = pg
  # plan_file = sqitch.plan
  # sql_dir = sql
  # deploy_dir = sql/deploy
  # revert_dir = sql/revert
  # test_dir = sql/test
  # extension = sql

# [core "pg"]
# db_name =
# client = psql
# sqitch_schema = sqitch
# password =
# port =
# host =
# username =
```

At the bottom of the window, the status bar displays the file name "sqitch.conf" and the mode "All (SQL[ansi])".

sqitch.conf

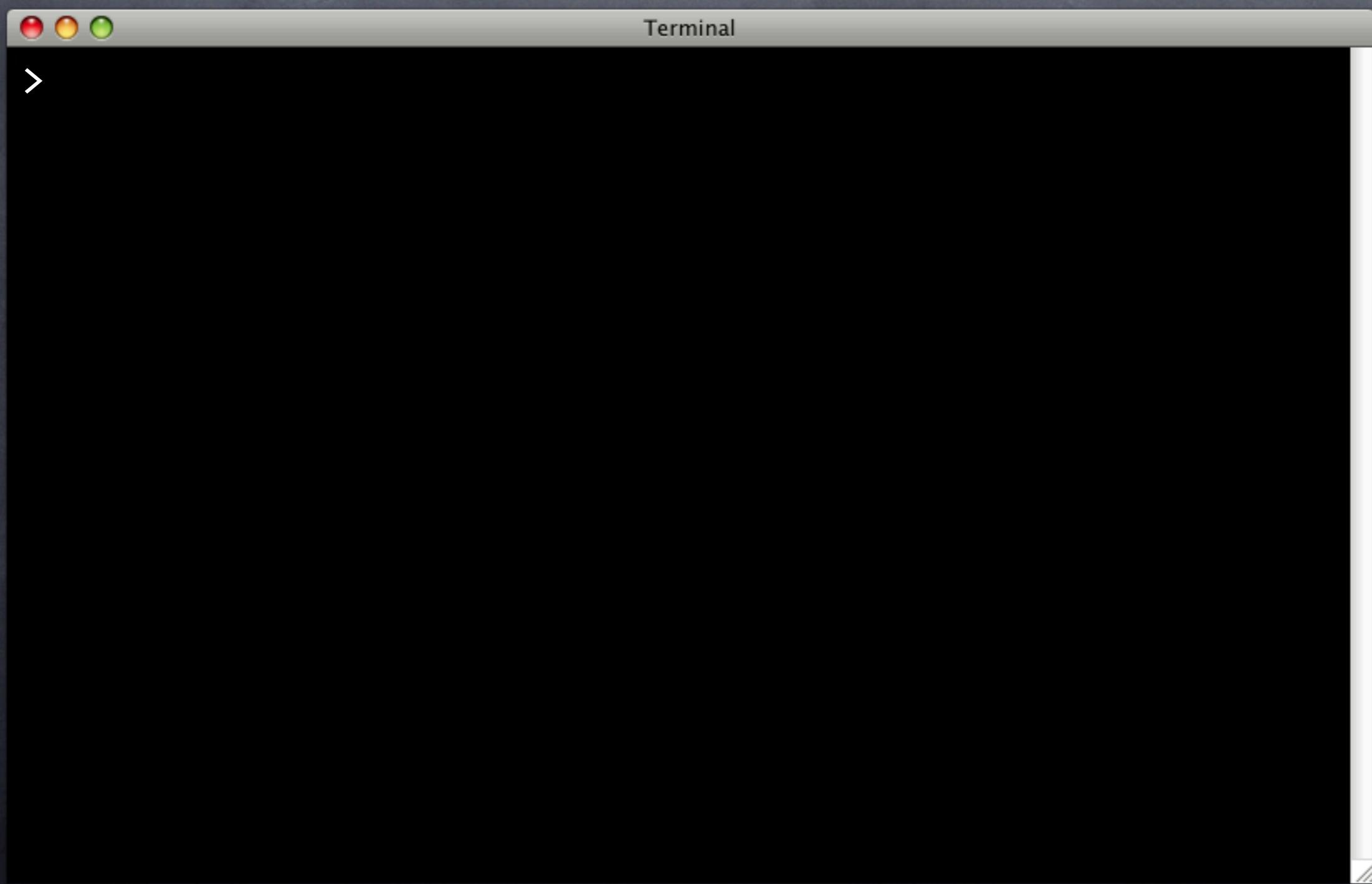


```
[core]
  engine = pg
  # plan_file =
  # sql_dir =
  # deploy_dir =
  # revert_dir =
  # test_dir =
  # extension =
# [core "pg"]
# db_name =
# client = psql
# sqitch_schema = sqitch
# password =
# port =
# host =
# username =
```

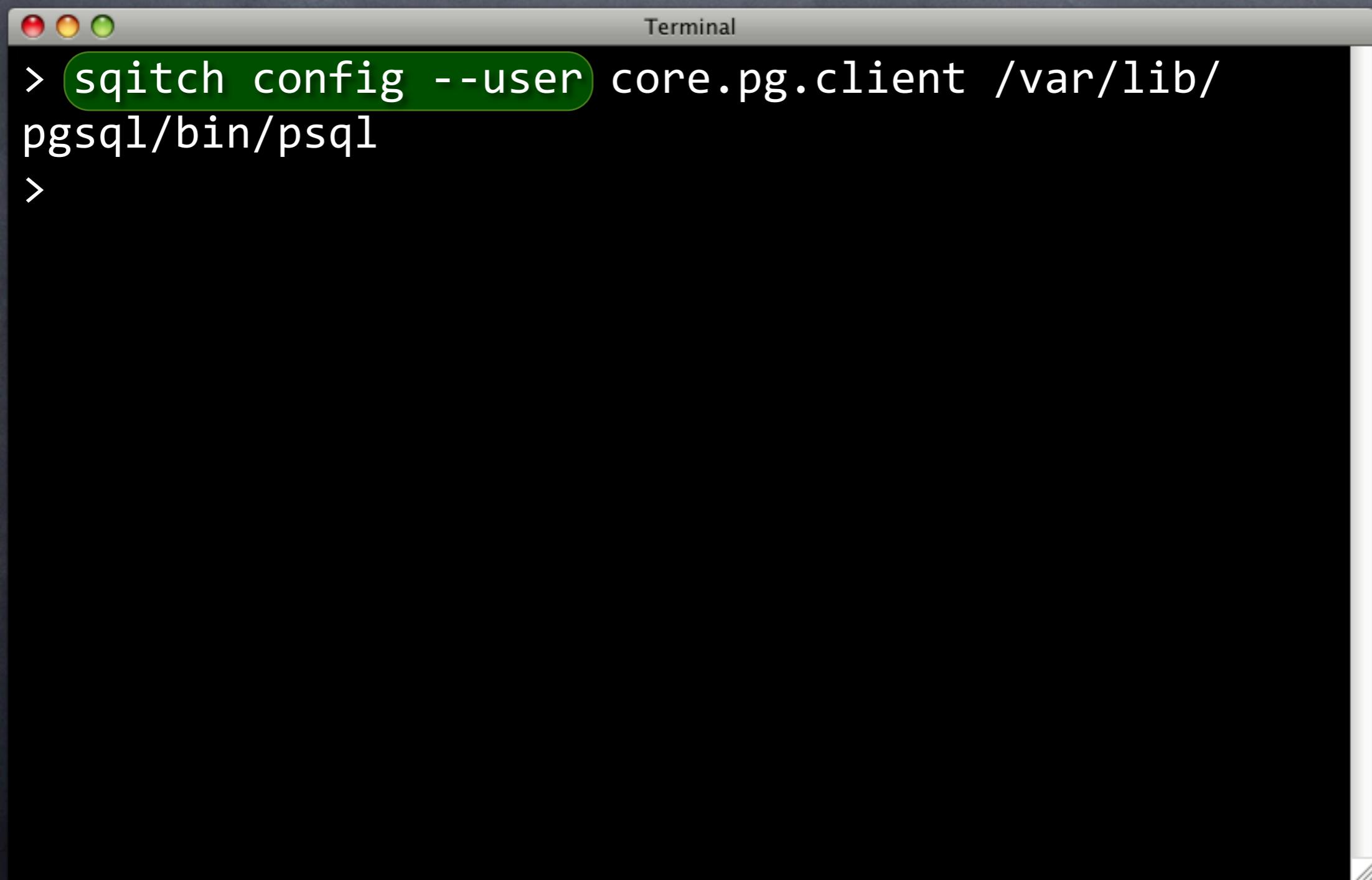
--engine pg

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

Add User Config



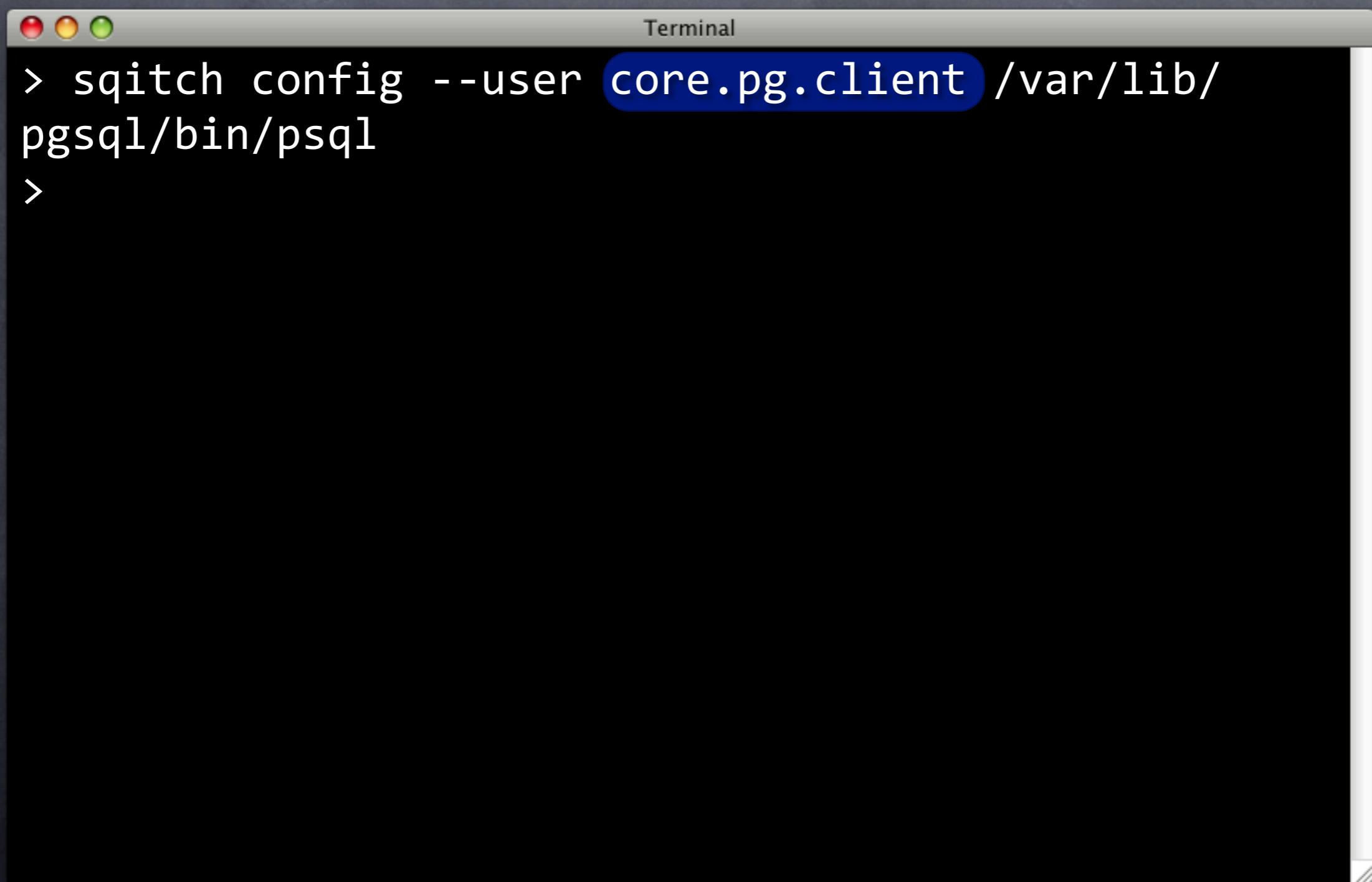
Add User Config



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 main area contains a command-line interface. A green rounded rectangle highlights the command "sqitch config --user core.pg.client /var/lib/pgsql/bin/psql". The command is preceded by a black right-pointing arrow. The rest of the terminal window is blank.

```
> sqitch config --user core.pg.client /var/lib/  
pgsql/bin/psql  
>
```

Add User Config

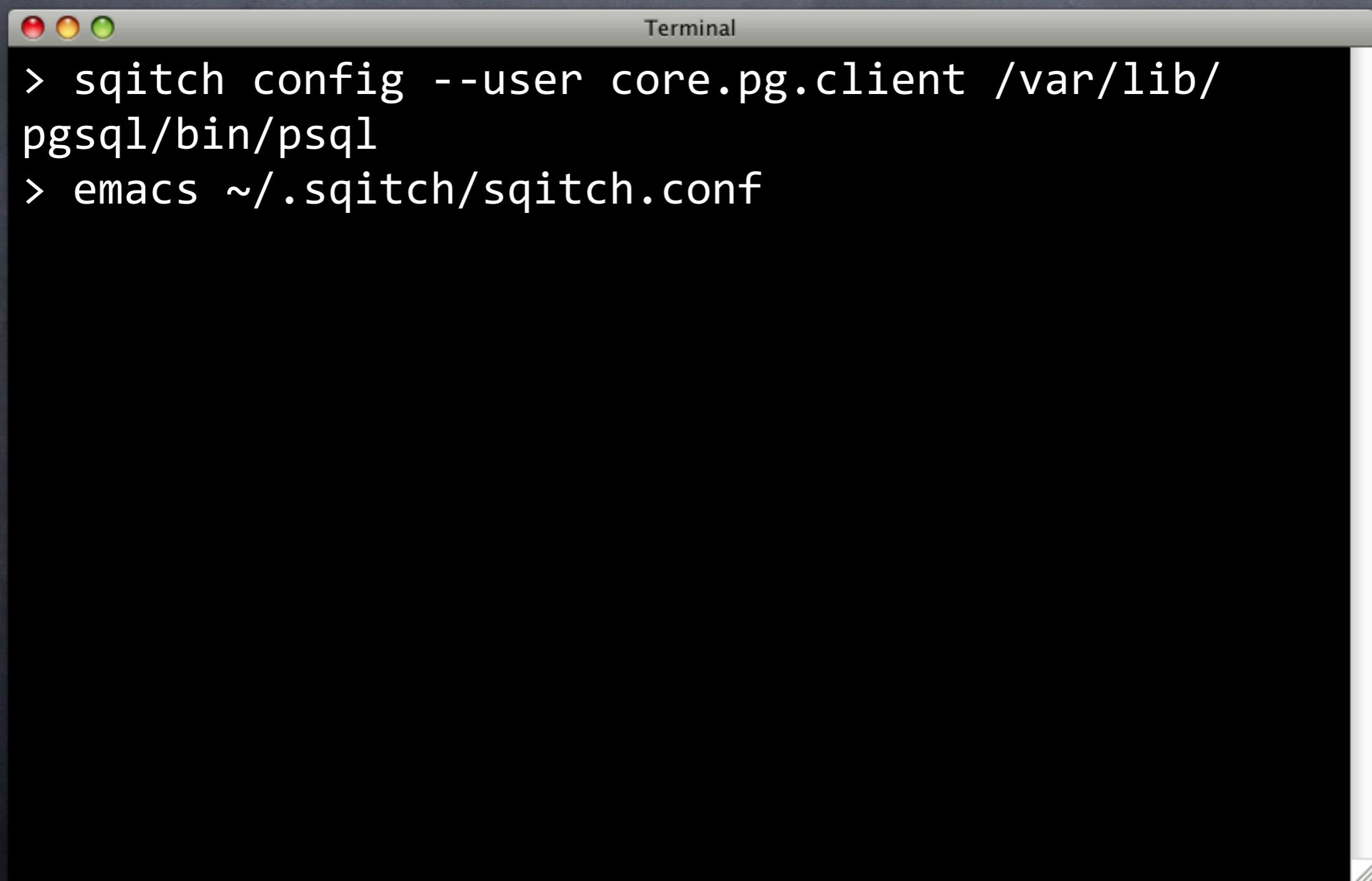


A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The main area contains a command-line interface. The command entered is:

```
> sqitch config --user core.pg.client /var/lib/  
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.

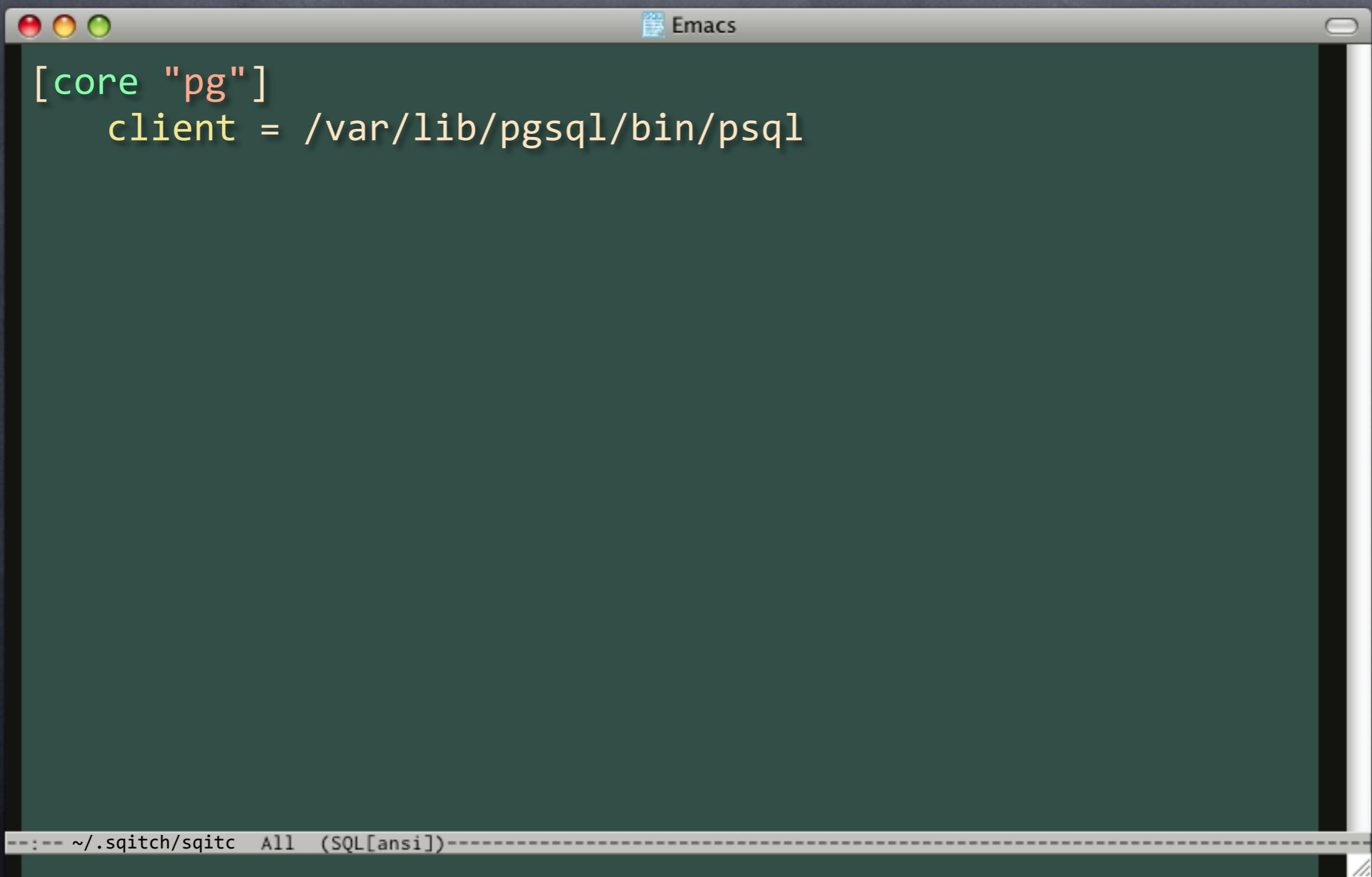
Add User Config



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 title bar reads "Terminal". The main pane contains two command-line entries:

```
> sqitch config --user core.pg.client /var/lib/
pgsql/bin/psql
> emacs ~/.sqitch/sqitch.conf
```

~/.sqitch/sqitch.conf

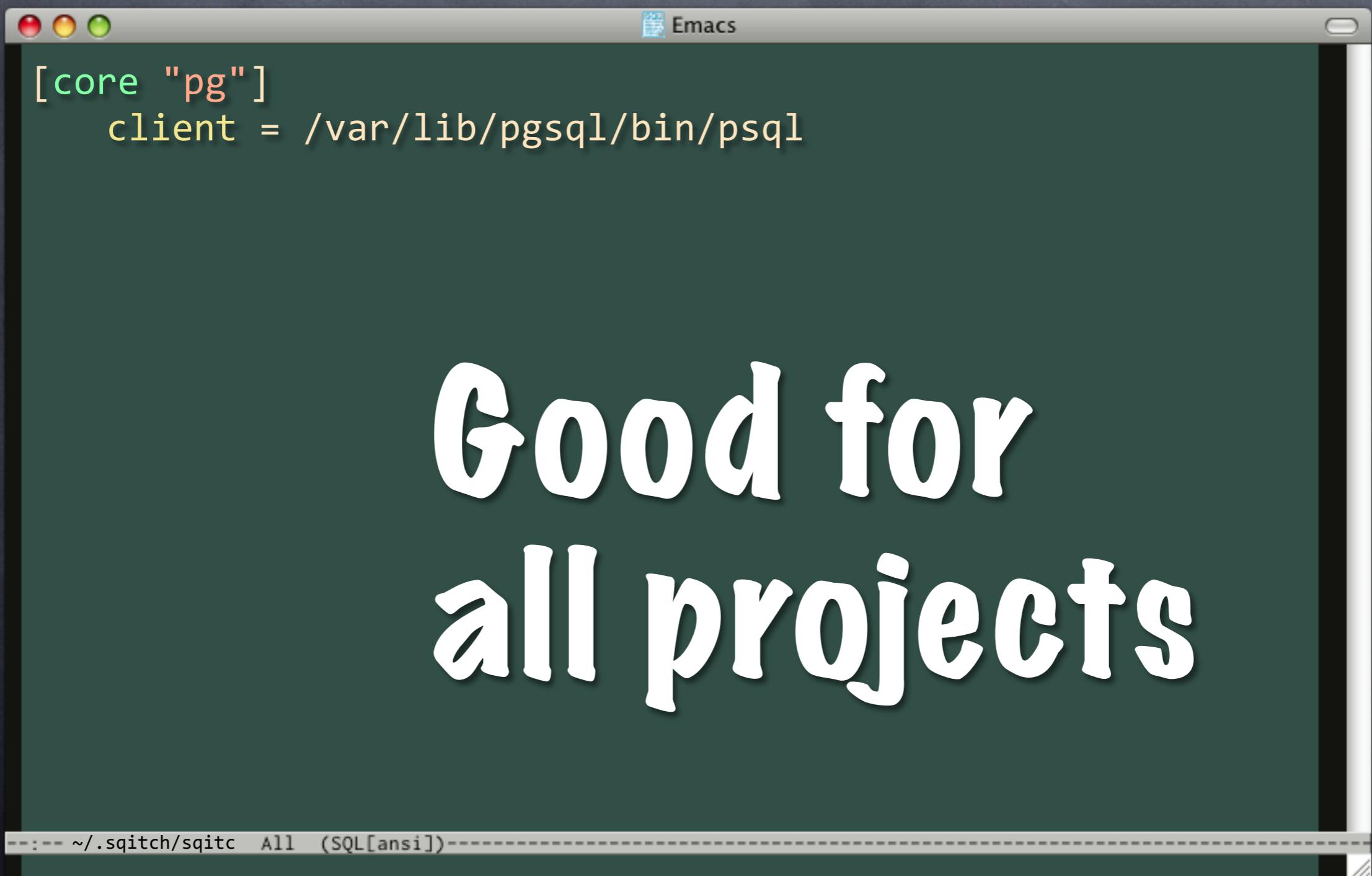


The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". The buffer contains the following configuration file:

```
[core "pg"]
client = /var/lib/pgsql/bin/psql
```

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

~/.sqitch/sqitch.conf

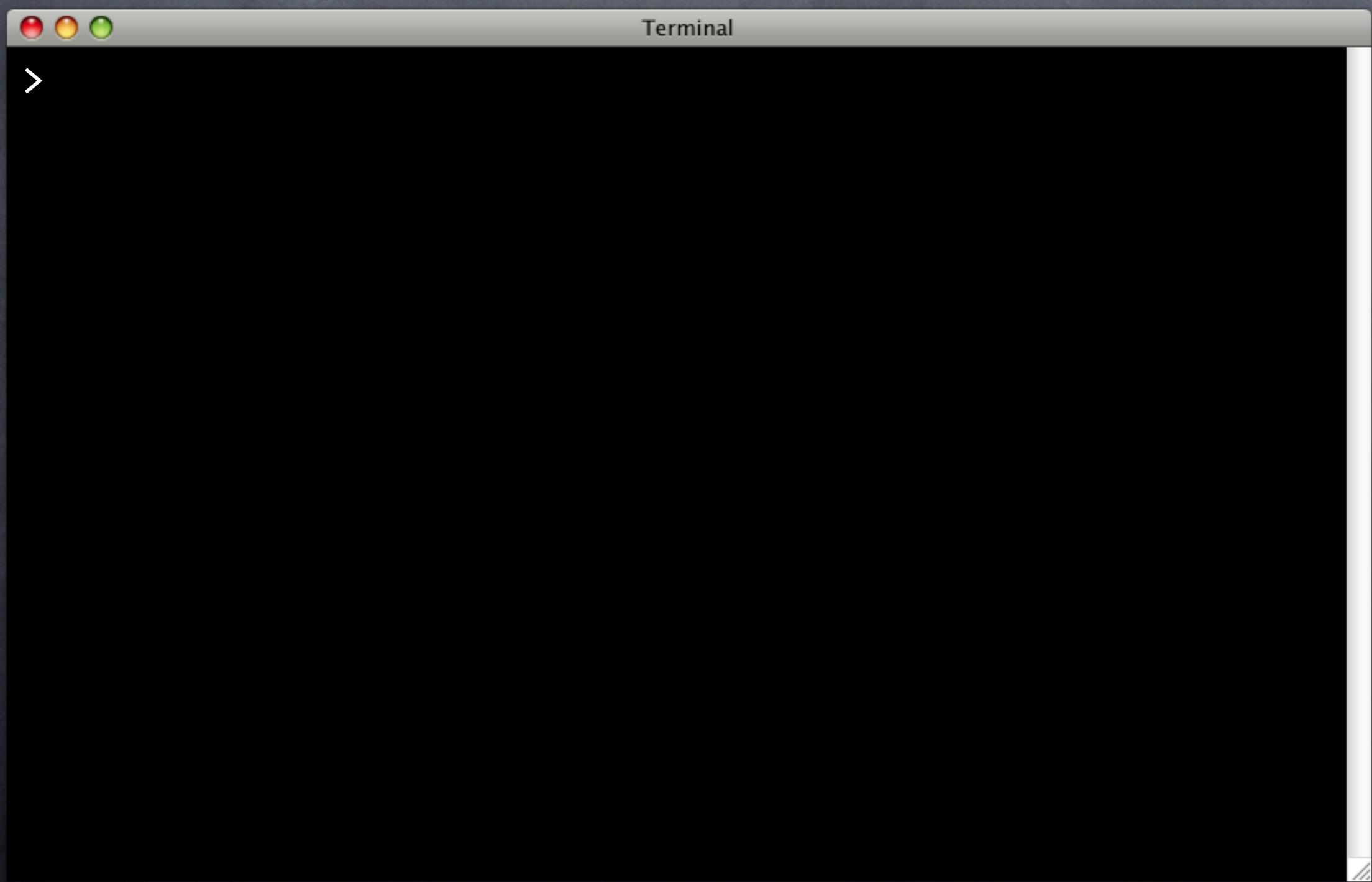


The image shows a screenshot of an Emacs window with a dark green background. The title bar reads "Emacs". Inside the window, there is a snippet of configuration code:

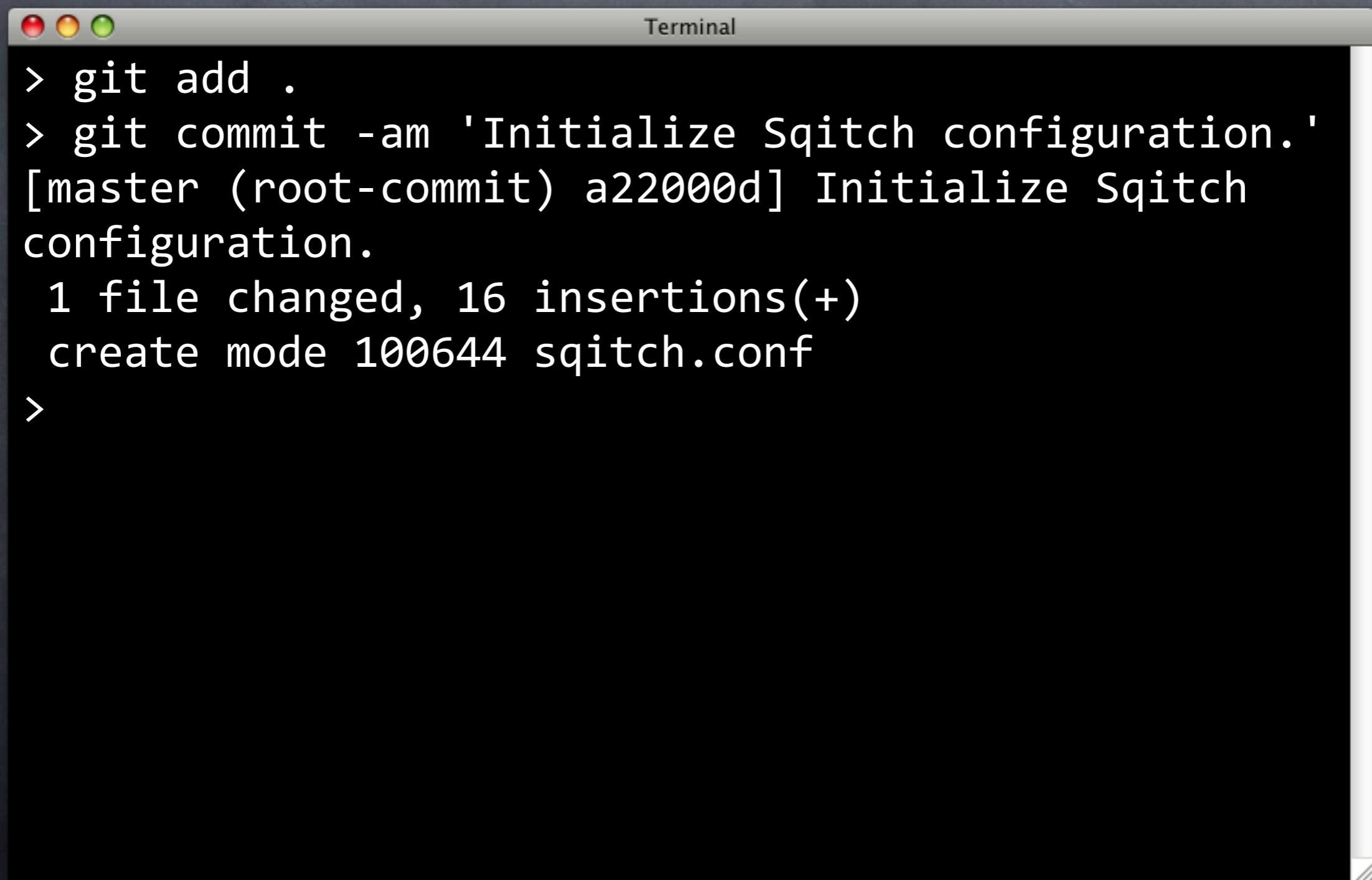
```
[core "pg"]
client = /var/lib/pgsql/bin/psql
```

Below the code, large white text reads "Good for all projects". At the bottom of the window, there is a status bar with the text "---- ~./sqitch/sqitc All (SQL[ansi])----".

Make It So

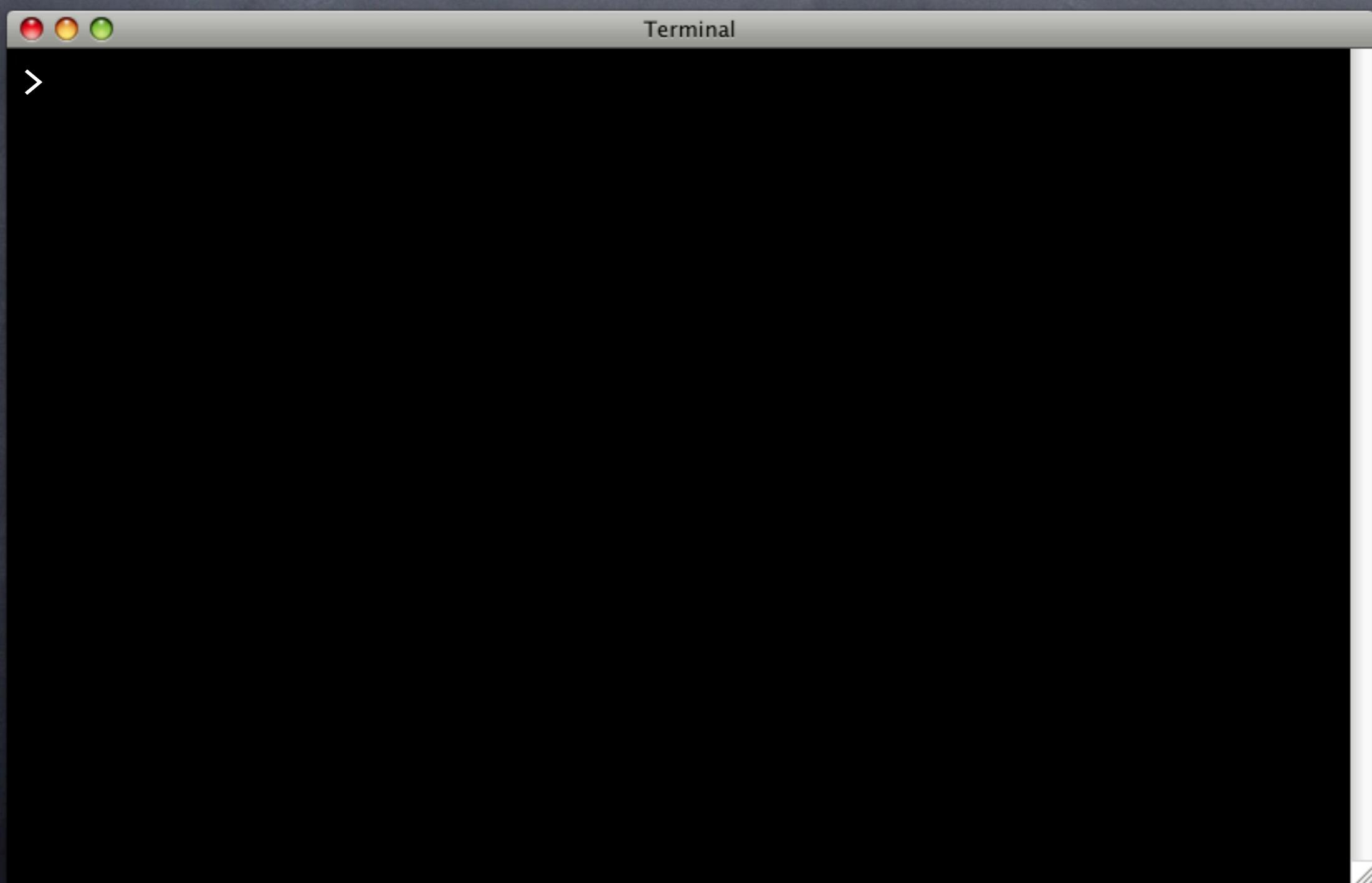


Make It So

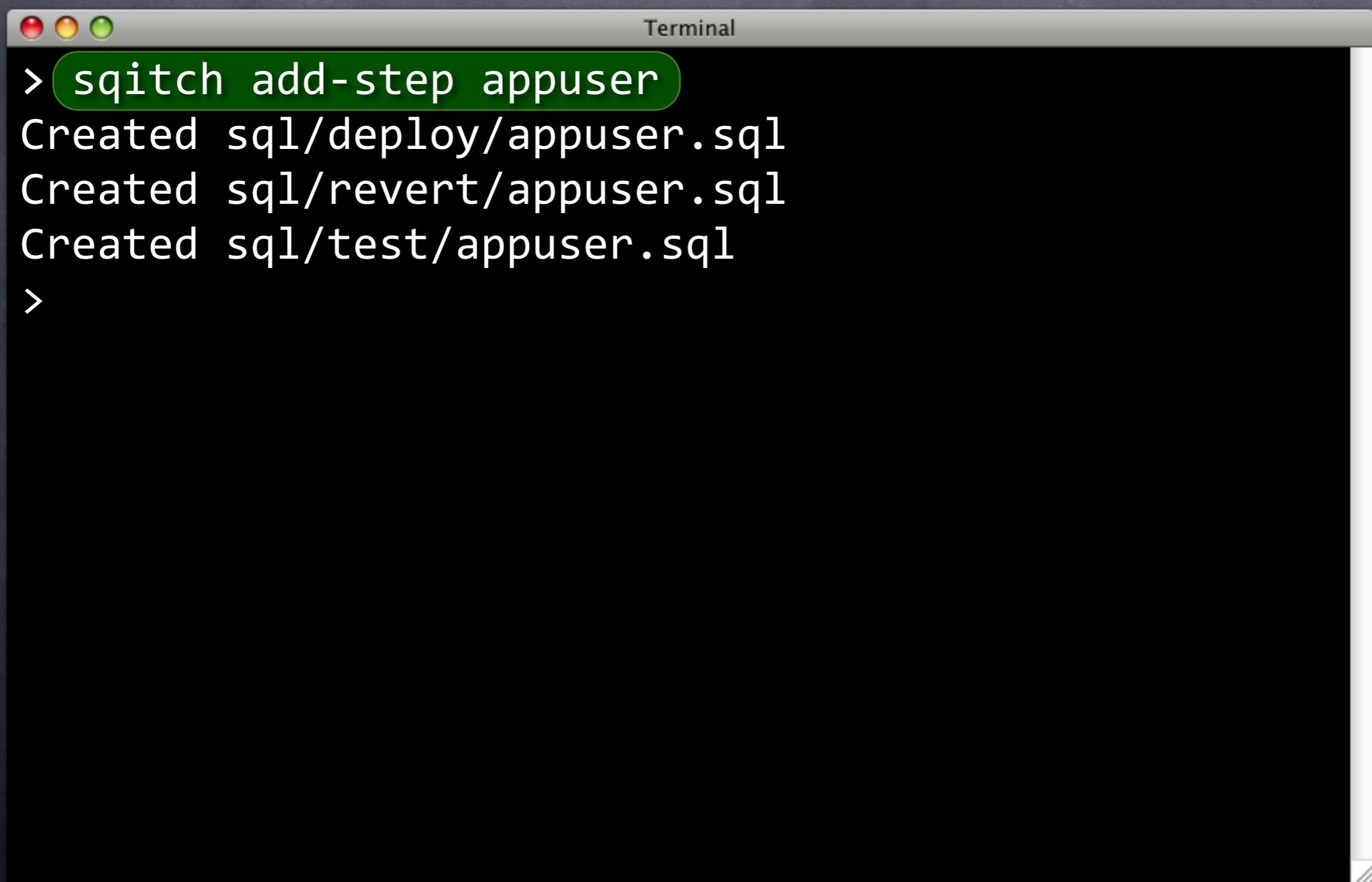


```
Terminal  
> git add .  
> git commit -am 'Initialize Sqitch configuration.'  
[master (root-commit) a22000d] Initialize Sqitch  
configuration.  
 1 file changed, 16 insertions(+)  
 create mode 100644 sqitch.conf  
>
```

First Deployment



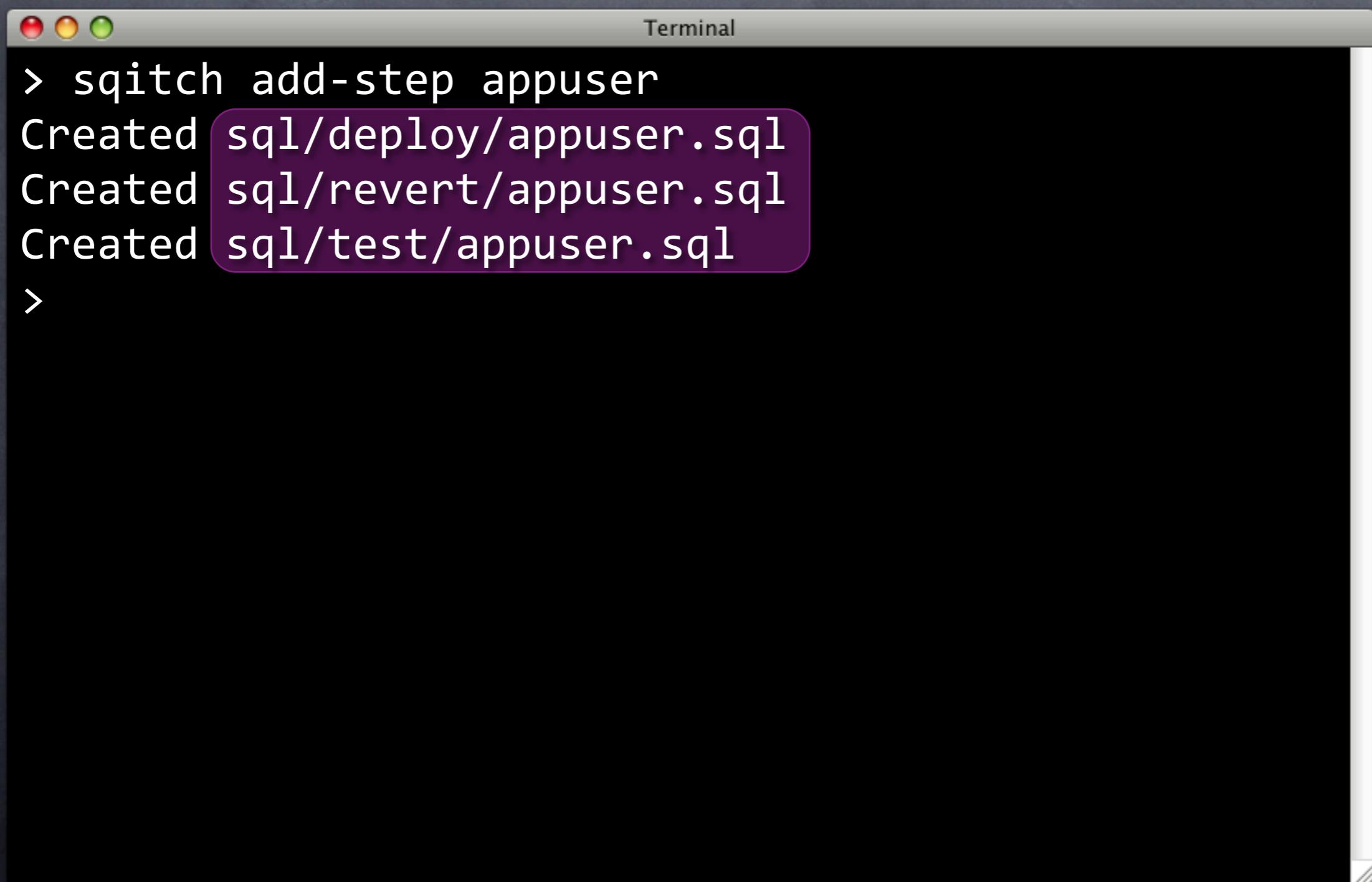
First Deployment

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-step appuser
Created sql/deploy/appuser.sql
Created sql/revert/appuser.sql
Created sql/test/appuser.sql
>
```

The text "sqitch add-step appuser" is highlighted with a green rounded rectangle.

First Deployment

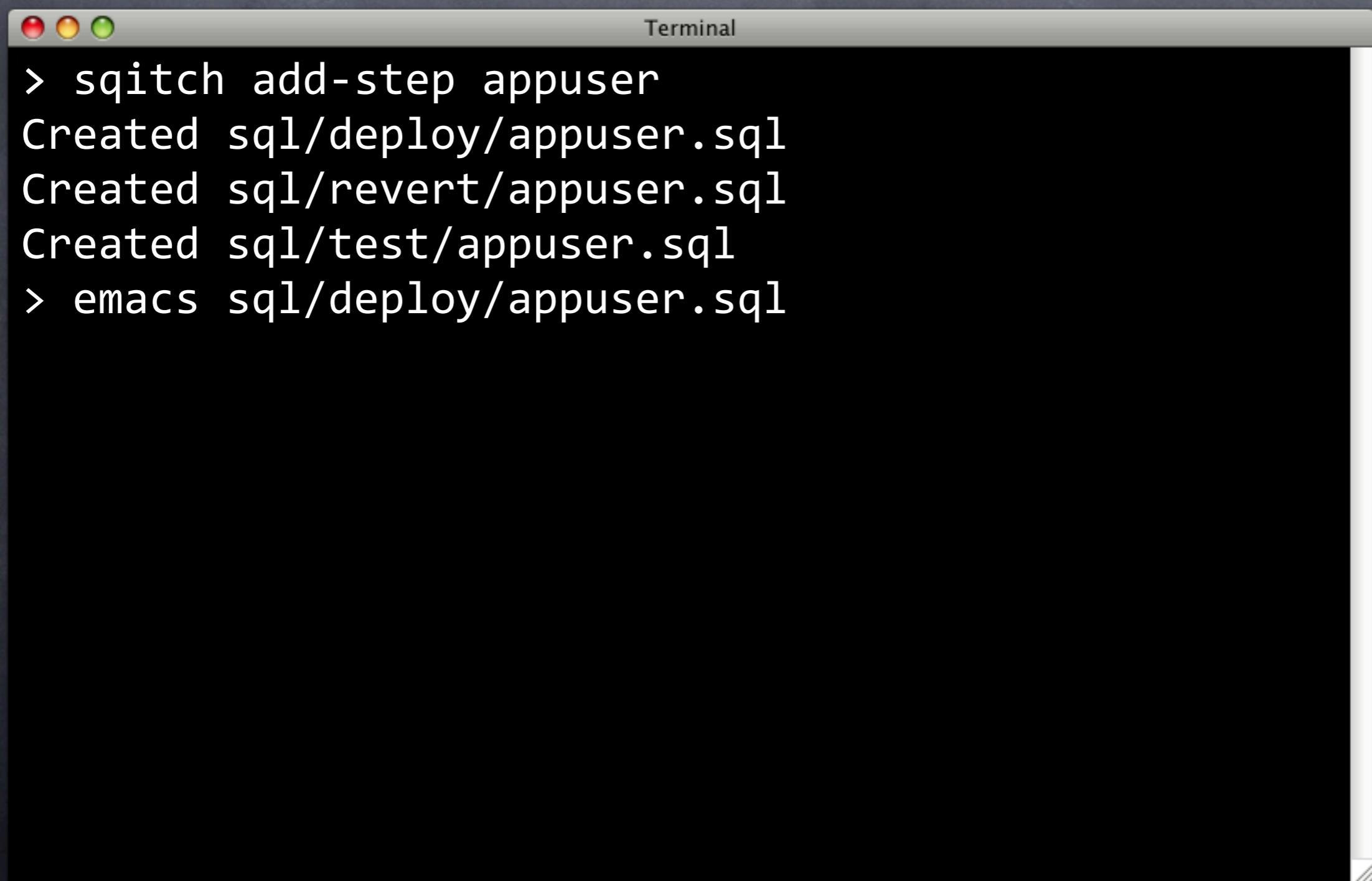


A screenshot of a Mac OS X Terminal window titled "Terminal". The window has the standard red, yellow, and green close buttons at the top left. The title bar reads "Terminal". The main pane contains the following text:

```
> sqitch add-step appuser
Created sql/deploy/appuser.sql
Created sql/revert/appuser.sql
Created sql/test/appuser.sql
>
```

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

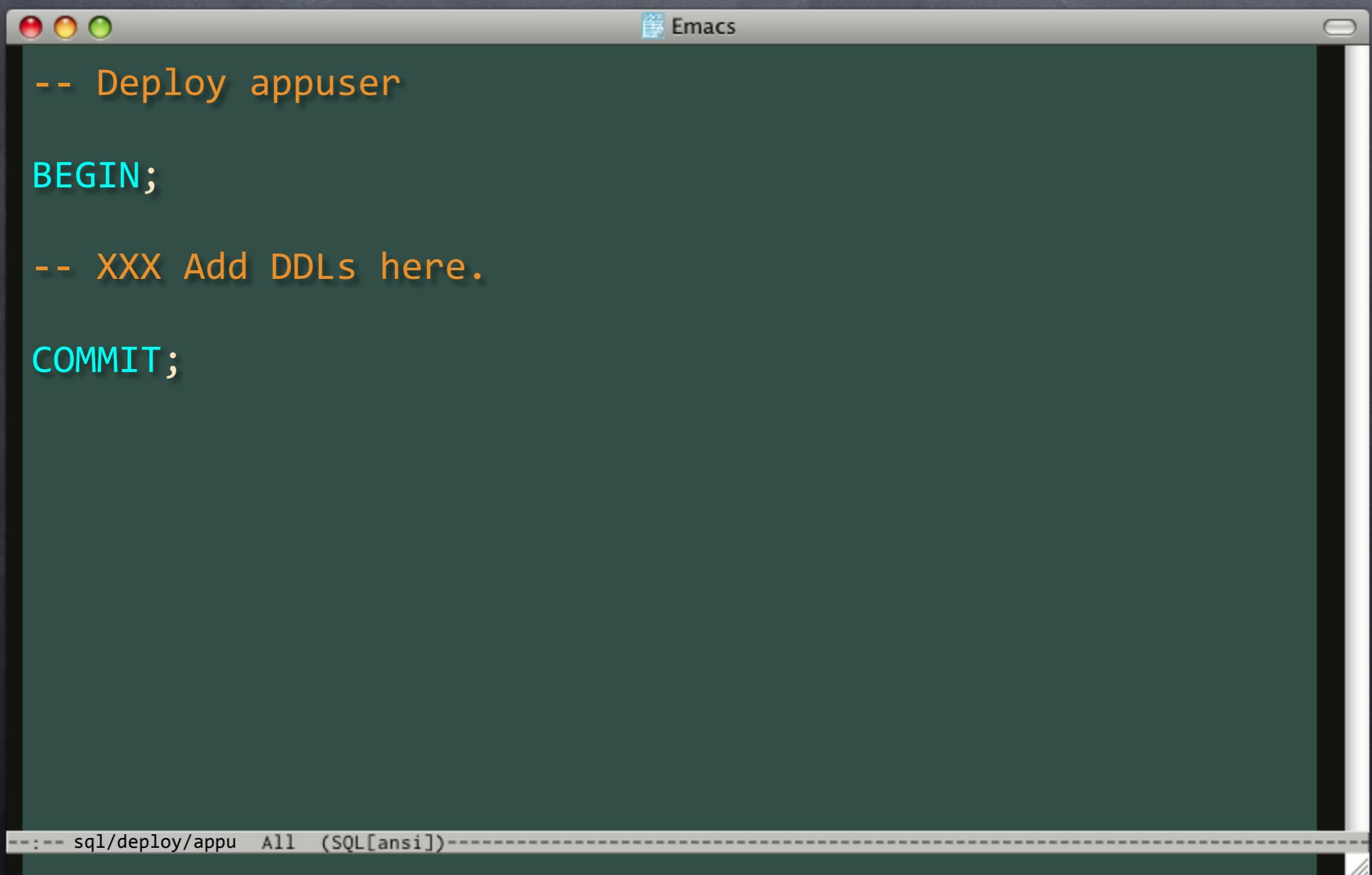
First Deployment



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-step appuser
Created sql/deploy/appuser.sql
Created sql/revert/appuser.sql
Created sql/test/appuser.sql
> emacs sql/deploy/appuser.sql
```

sql/deploy/appuser.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 deployment script:

```
-- Deploy appuser

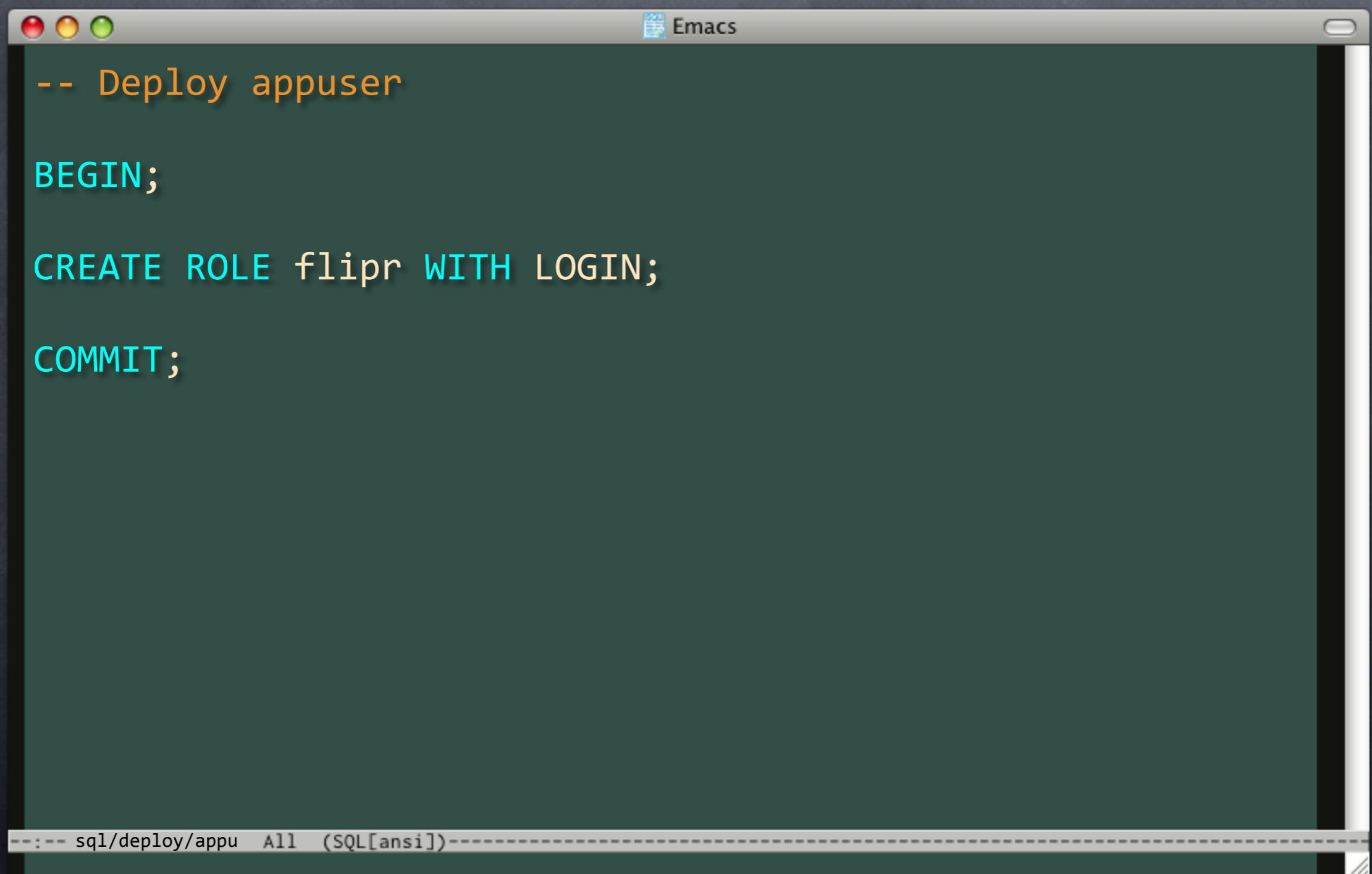
BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

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

sql/deploy/appuser.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 script:

```
-- Deploy appuser

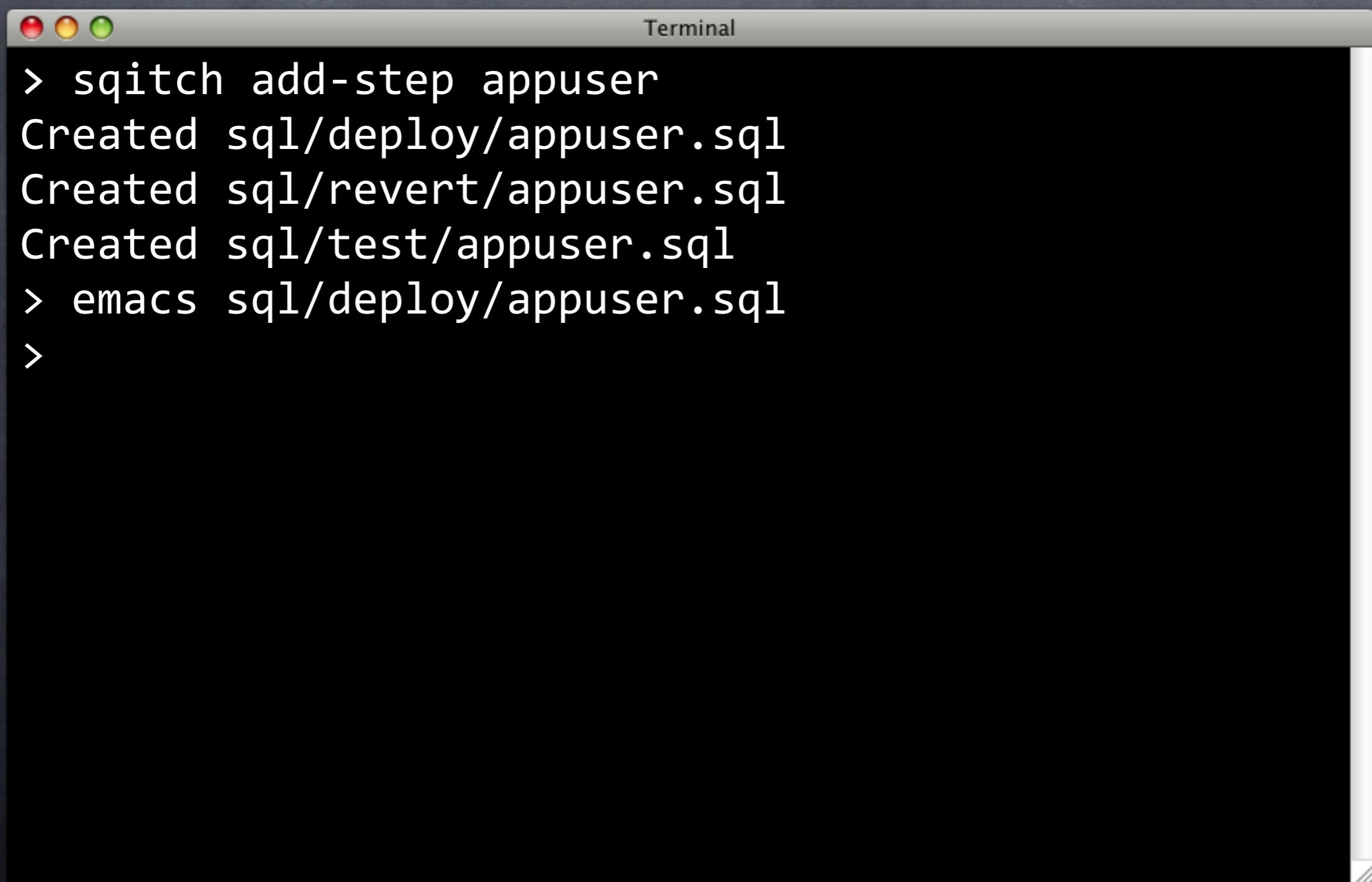
BEGIN;

CREATE ROLE flipr WITH LOGIN;

COMMIT;
```

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

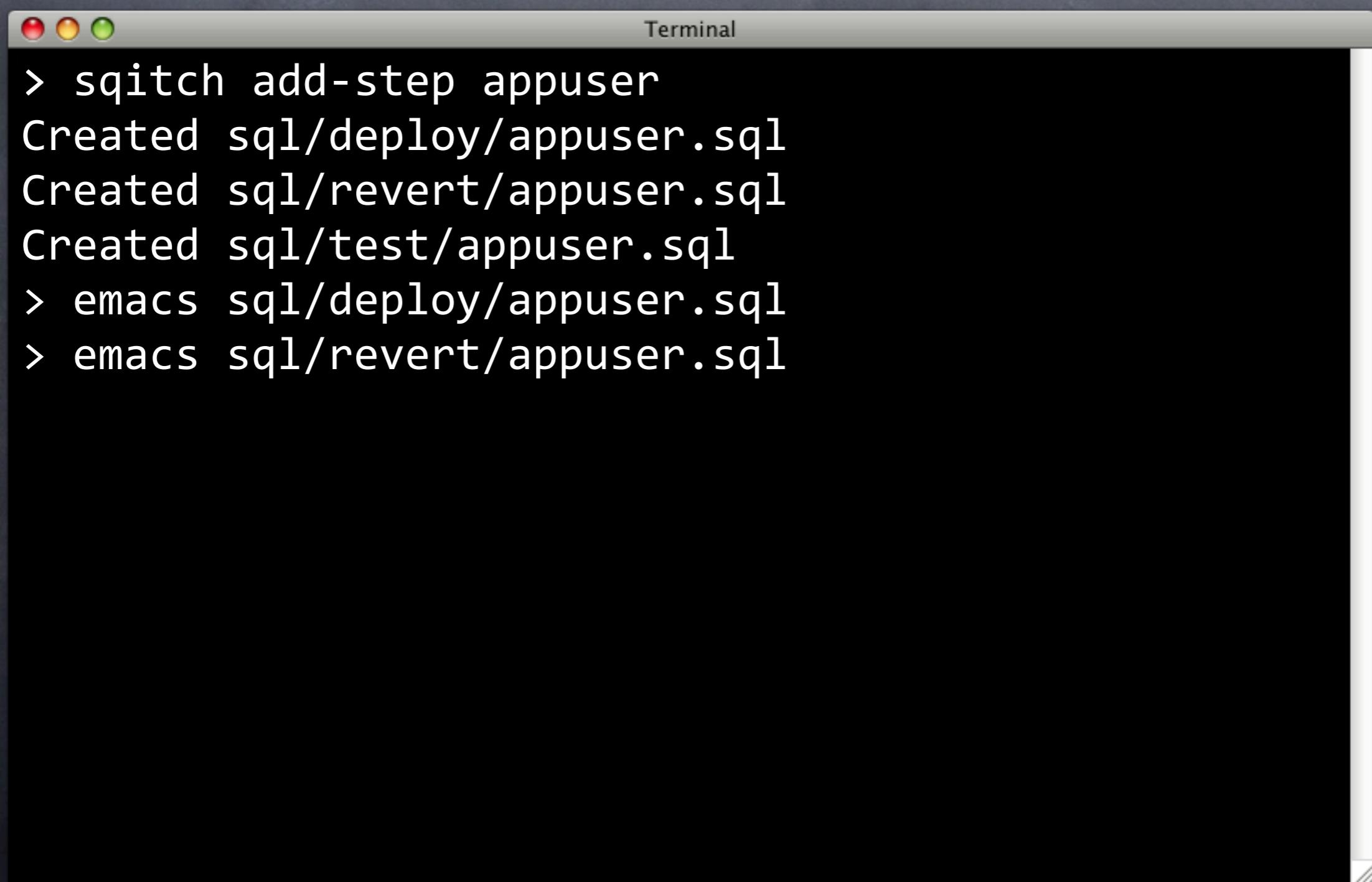
First Deployment



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-step appuser
Created sql/deploy/appuser.sql
Created sql/revert/appuser.sql
Created sql/test/appuser.sql
> emacs sql/deploy/appuser.sql
>
```

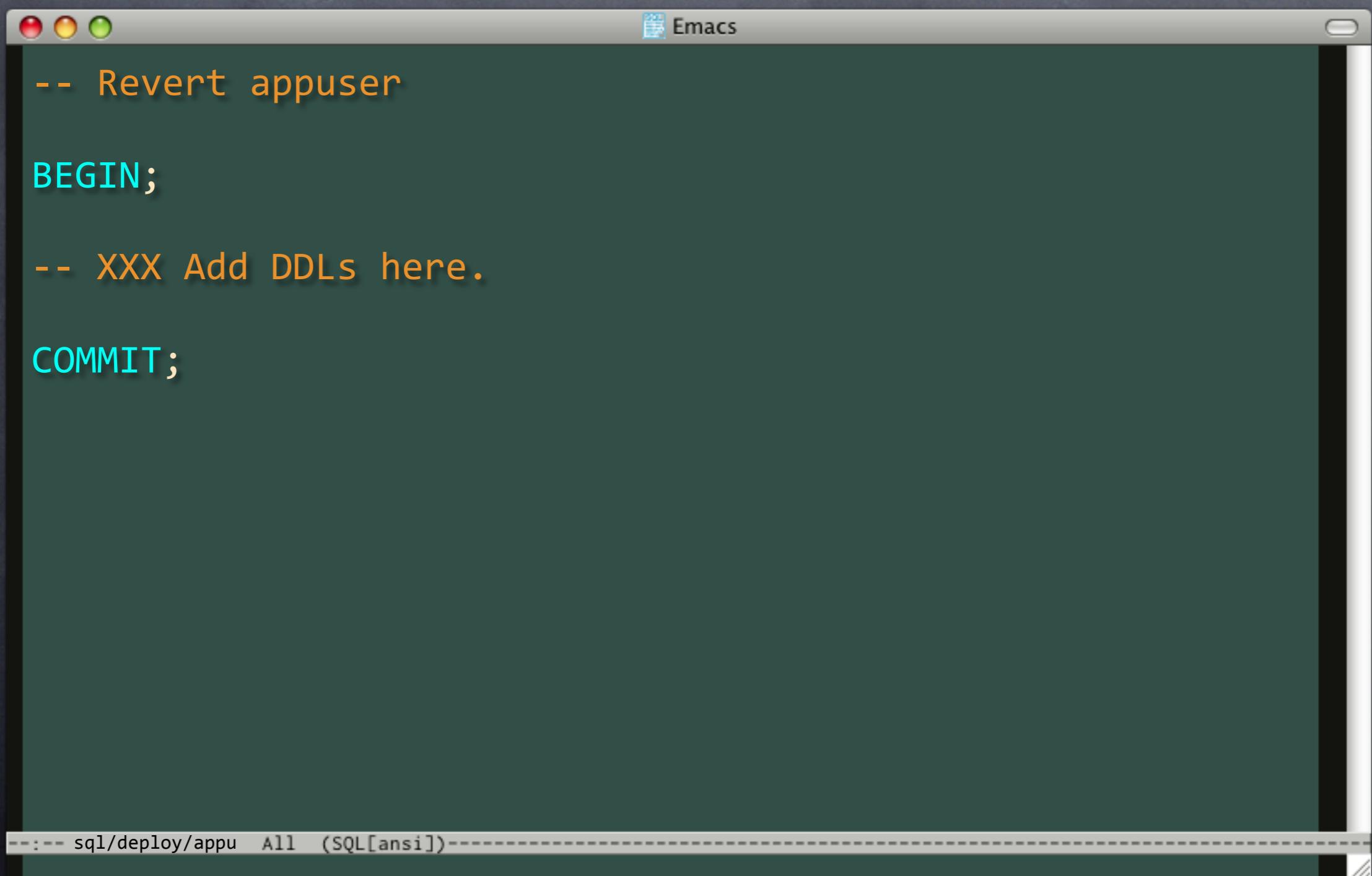
First Deployment



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-step appuser
Created sql/deploy/appuser.sql
Created sql/revert/appuser.sql
Created sql/test/appuser.sql
> emacs sql/deploy/appuser.sql
> emacs sql/revert/appuser.sql
```

sql/revert/appuser.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:

```
-- Revert appuser

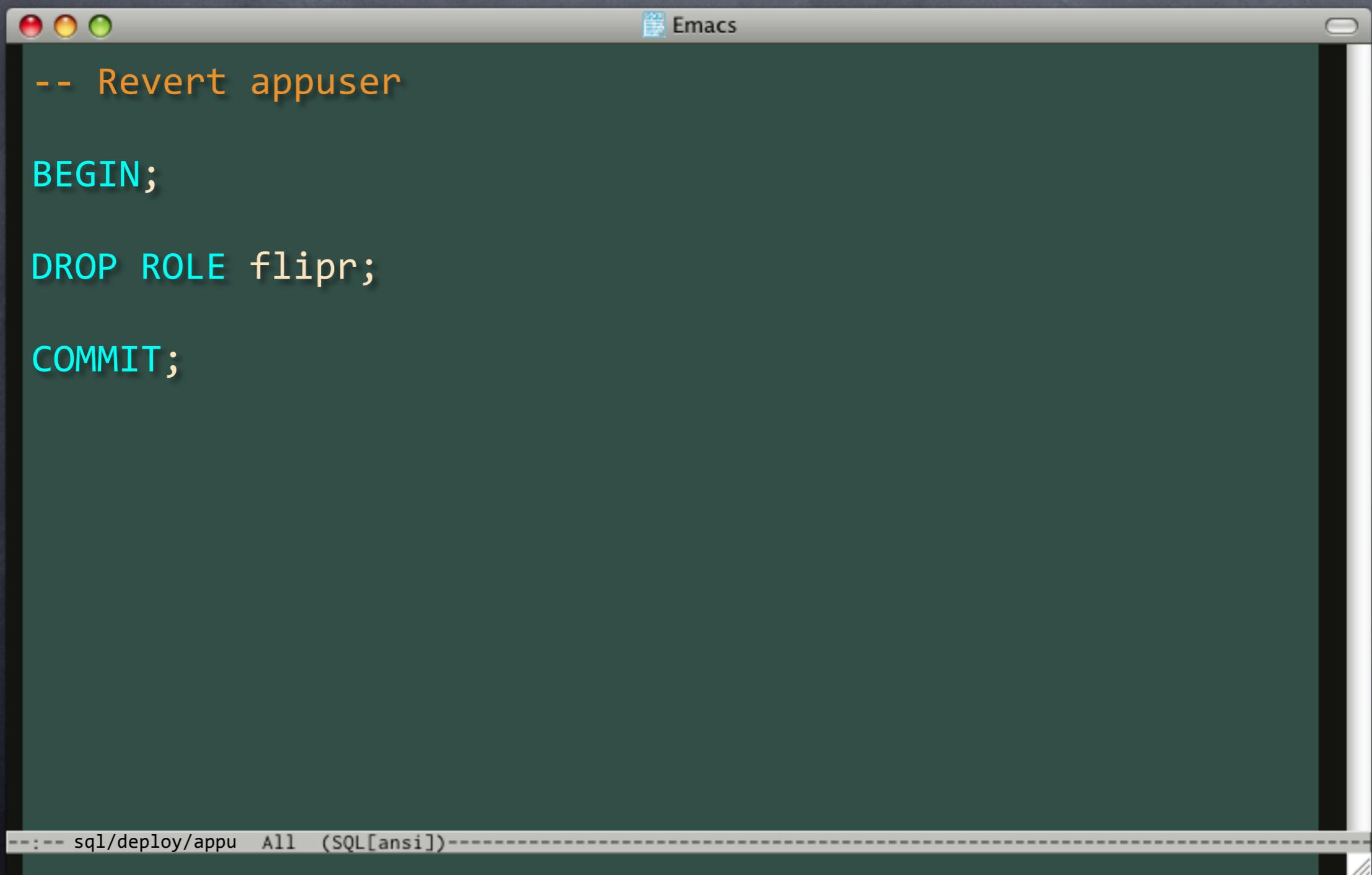
BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

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

sql/revert/appuser.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:

```
-- Revert appuser

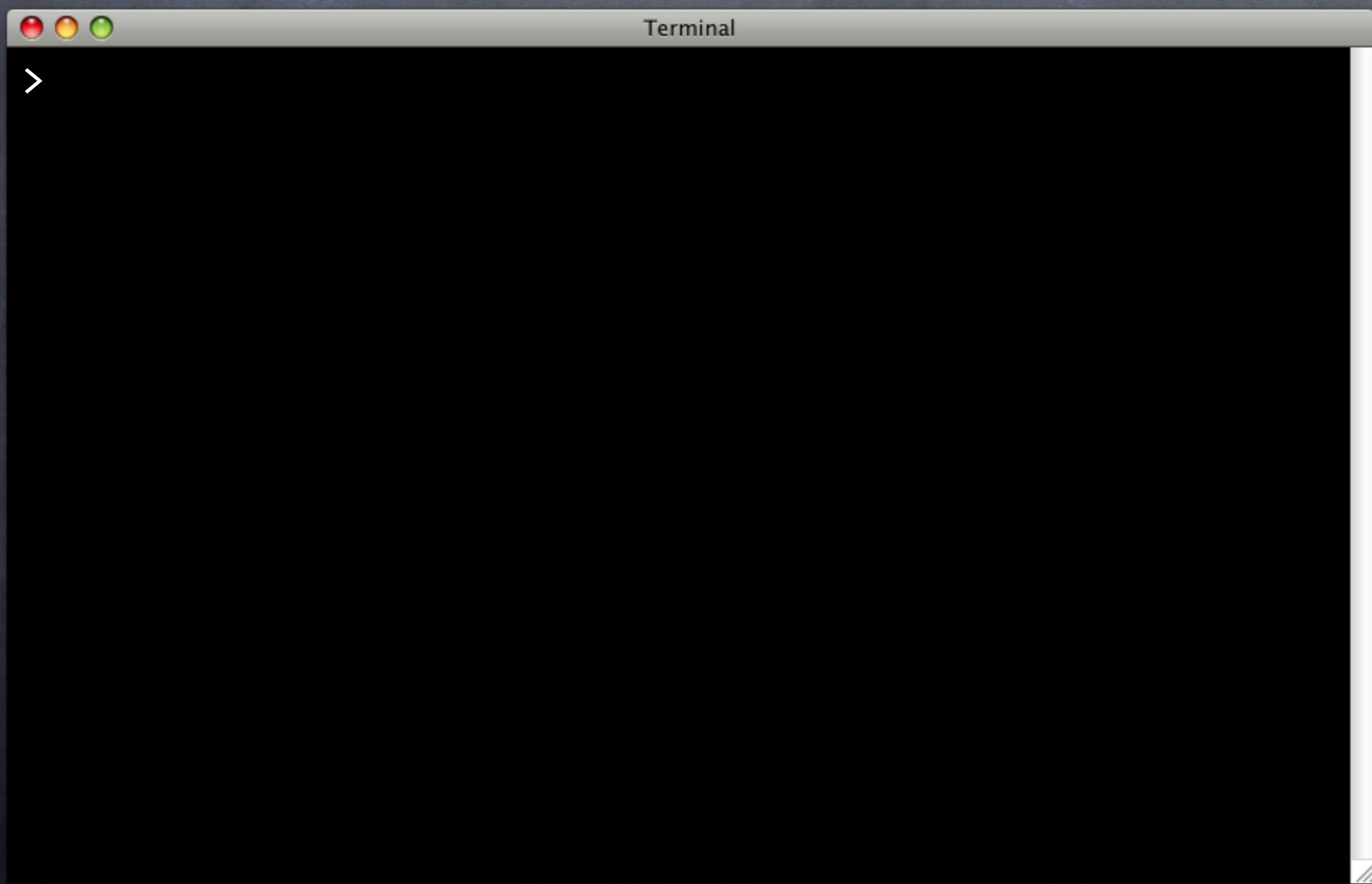
BEGIN;

DROP ROLE flipr;

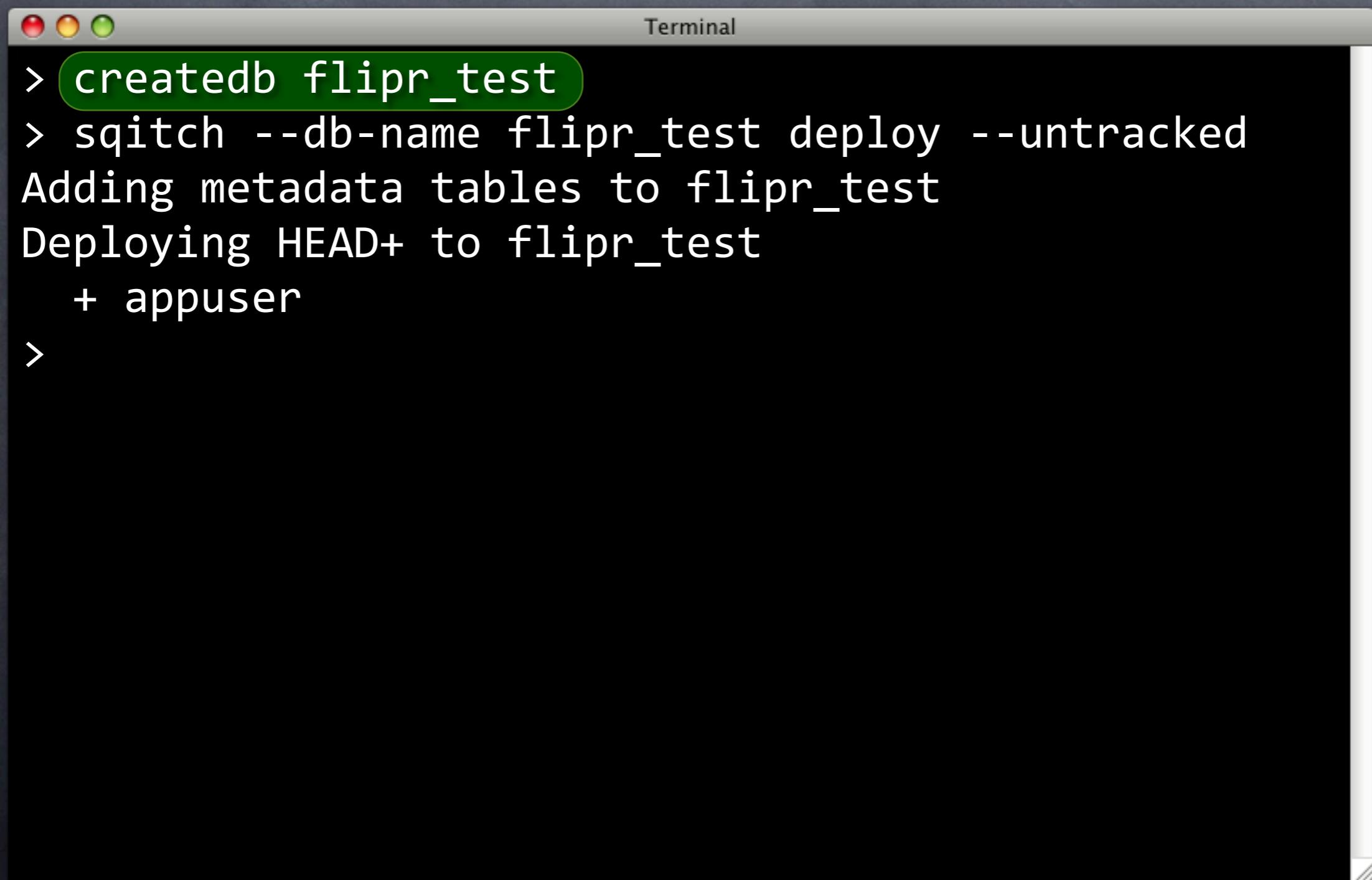
COMMIT;
```

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

Make it So!



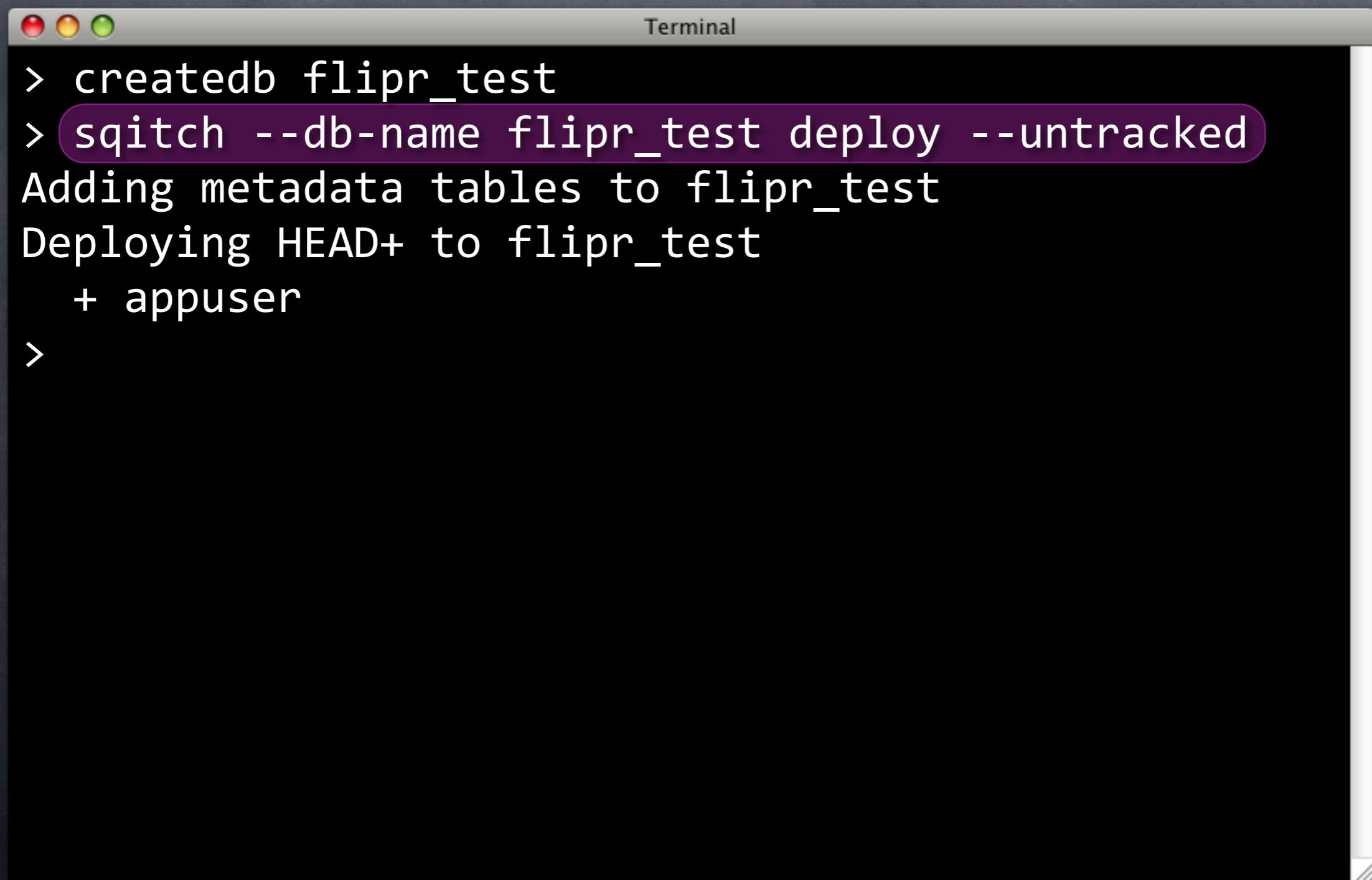
Make it So!



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 has a black background with white text. A command is highlighted with a green rounded rectangle. The text in the terminal is as follows:

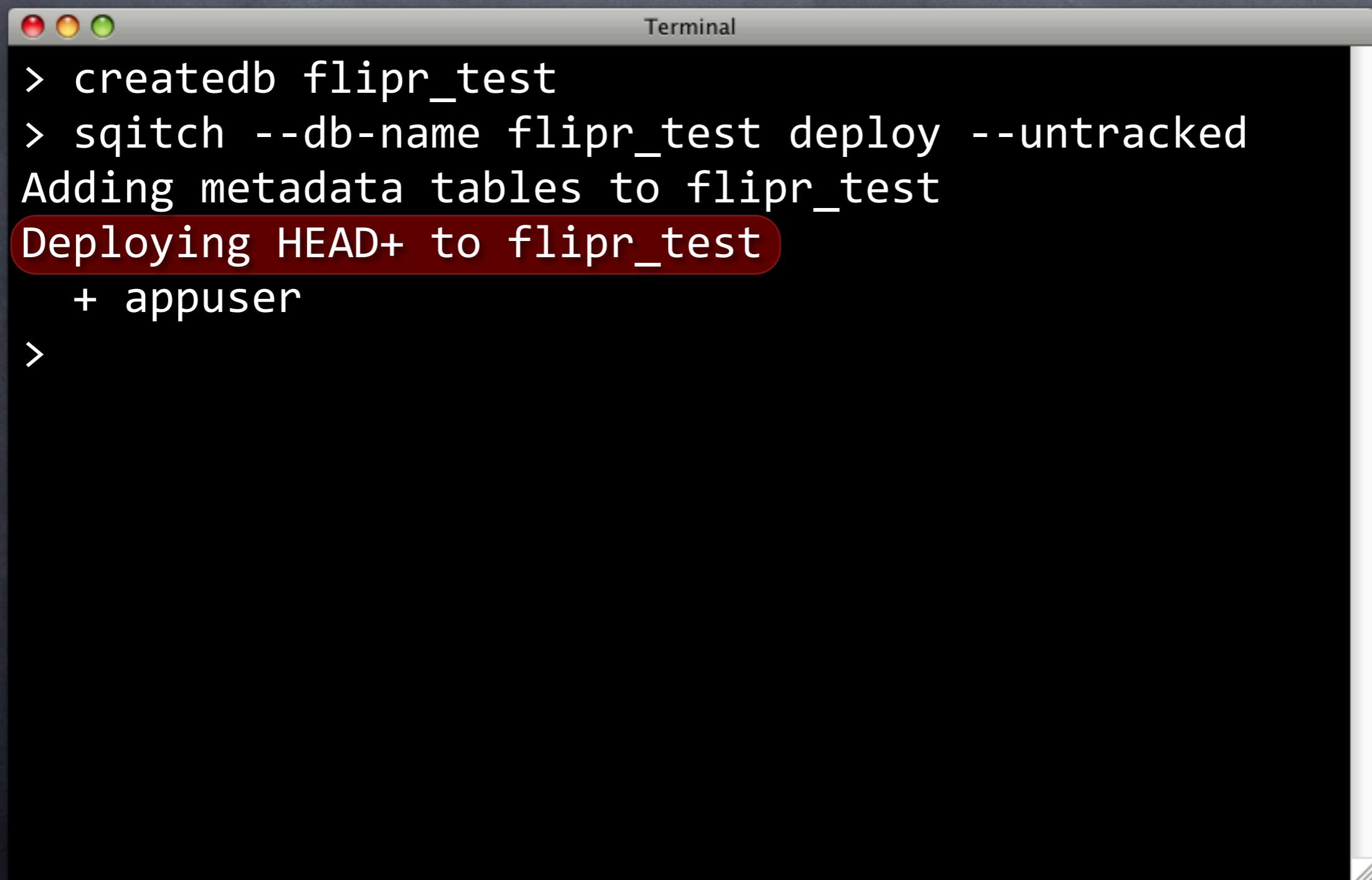
```
> createdb flipr_test
> sqitch --db-name flipr_test deploy --untracked
Adding metadata tables to flipr_test
Deploying HEAD+ to flipr_test
  + appuser
>
```

Make it So!



```
Terminal
> createdb flipr_test
> sqitch --db-name flipr_test deploy --untracked
Adding metadata tables to flipr_test
Deploying HEAD+ to flipr_test
  + appuser
>
```

Make it So!

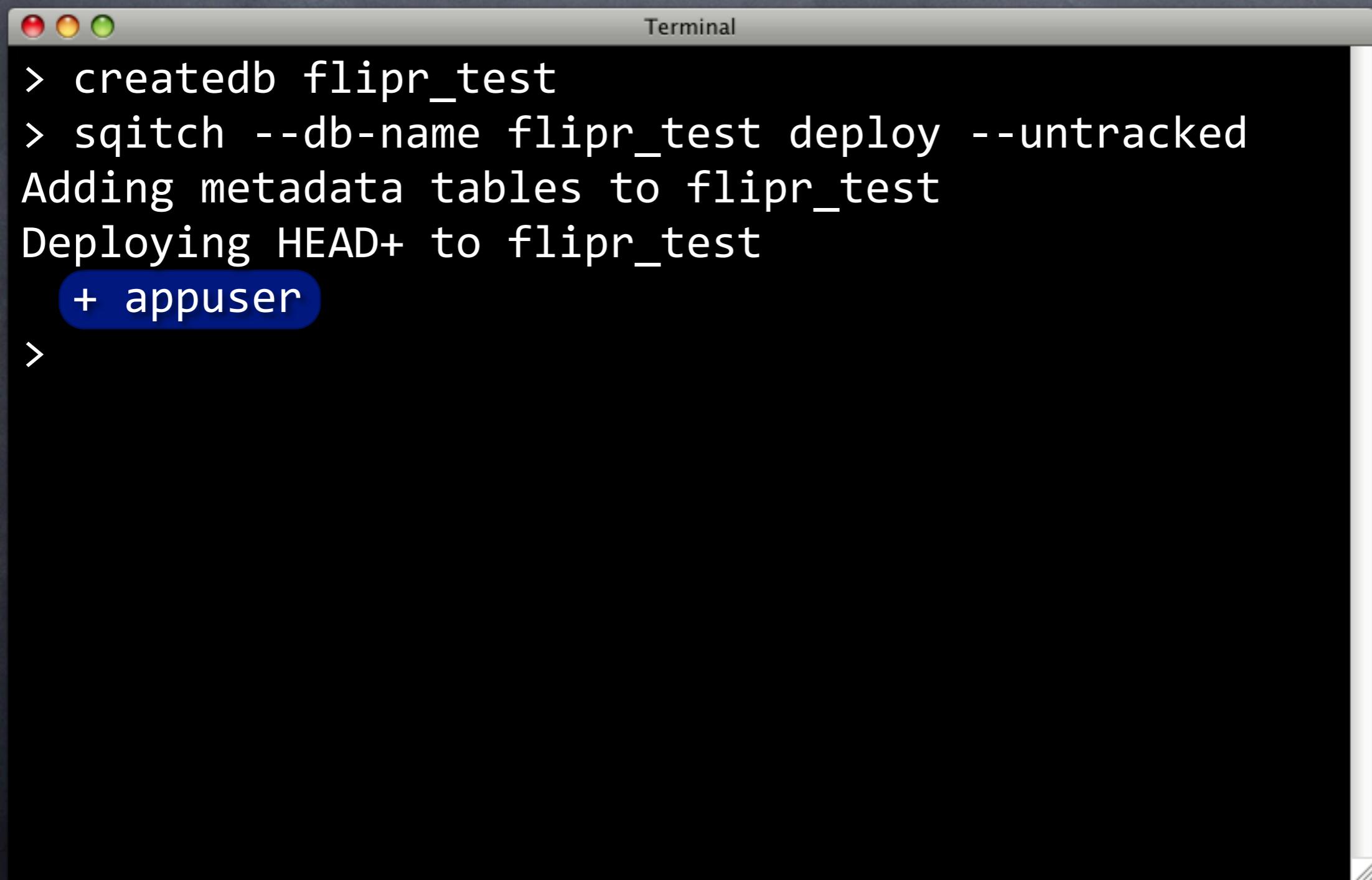


The image shows a Mac OS X terminal window titled "Terminal". The window has the classic red, yellow, and green close buttons at the top left. The terminal itself has a dark background with white text. It displays the following command-line session:

```
> createdb flipr_test
> sqitch --db-name flipr_test deploy --untracked
Adding metadata tables to flipr_test
Deploying HEAD+ to flipr_test
+ appuser
>
```

The line "Deploying HEAD+ 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 has the standard red, yellow, and green close buttons at the top left. The terminal itself is black with white text. It shows the following command-line session:

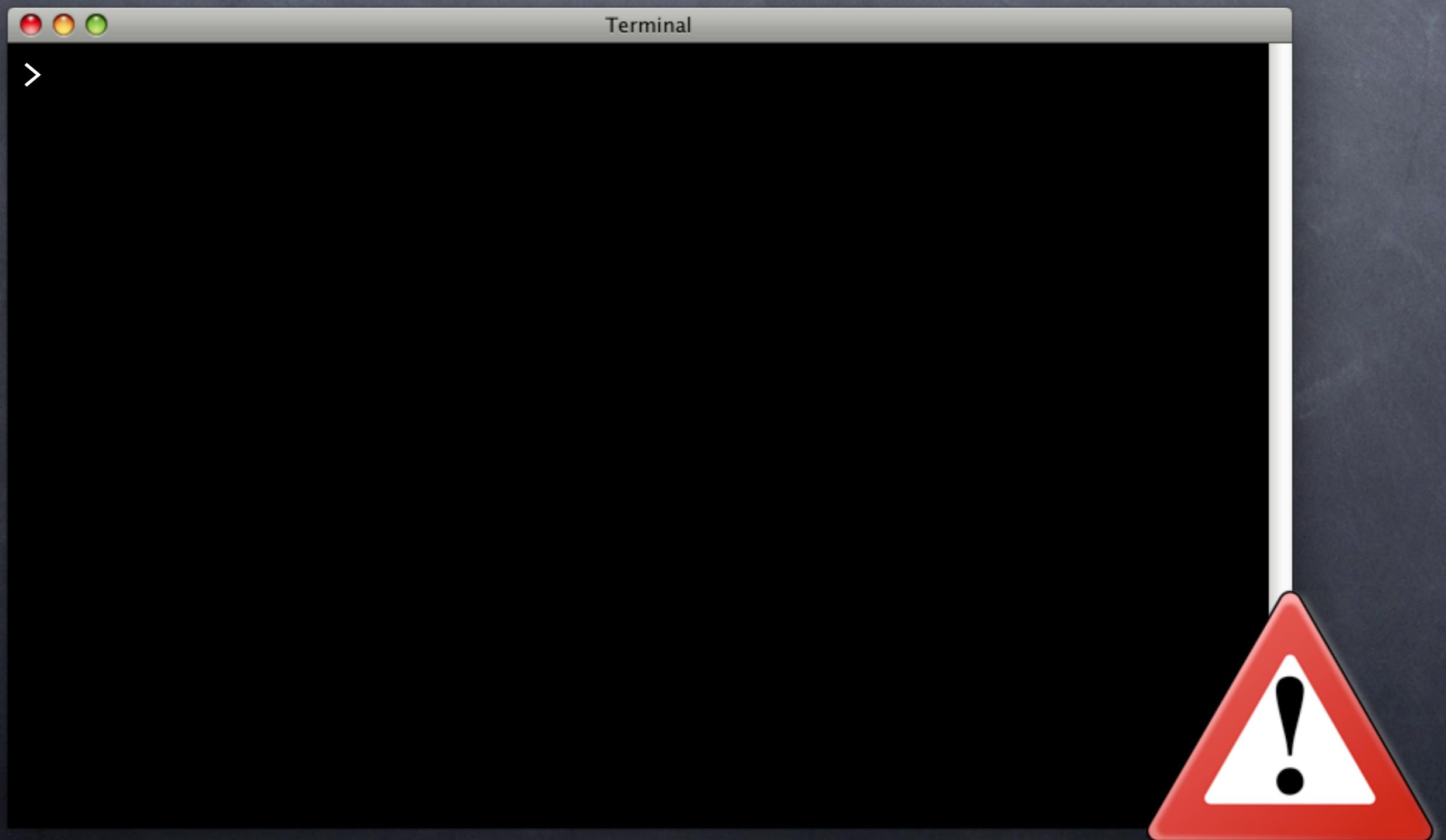
```
> createdb flipr_test
> sqitch --db-name flipr_test deploy --untracked
Adding metadata tables to flipr_test
Deploying HEAD+ to flipr_test
+ appuser
>
```

The text "+ appuser" is highlighted with a blue rounded rectangle.

Make it So!

```
Terminal  
> createdb flipr_test  
> sqitch --db-name flipr_test deploy --untracked  
Adding metadata tables to flipr_test  
Deploying HEAD+ to flipr_test  
  + appuser  
> psql -d flipr_test -c '\du flipr'  
      List of roles  
 Role name | Attributes | Member of  
-----+-----+-----  
 flipr | | {}  
>
```

How's it Look?



How's it Look?

```
Terminal  
> sqitch -d flipr_test status  
# On database flipr_test  
# Tag: HEAD+  
# Step: appuser  
# Date: 2012-05-18 17:15:23  
#  
Nothing to deploy (up-to-date)  
>
```



How's it Look?

```
Terminal  
> sqitch -d flipr_test status  
# On database flipr_test  
# Tag: HEAD+  
# Step: appuser  
# Date: 2012-05-18 17:15:23  
#  
Nothing to deploy (up-to-date)  
>
```



How's it Look?

```
Terminal  
> sqitch -d flipr_test status  
# On database flipr_test  
# Tag: HEAD+  
# Step: appuser  
# Date: 2012-05-18 17:15:23  
#  
Nothing to deploy (up-to-date)  
>
```



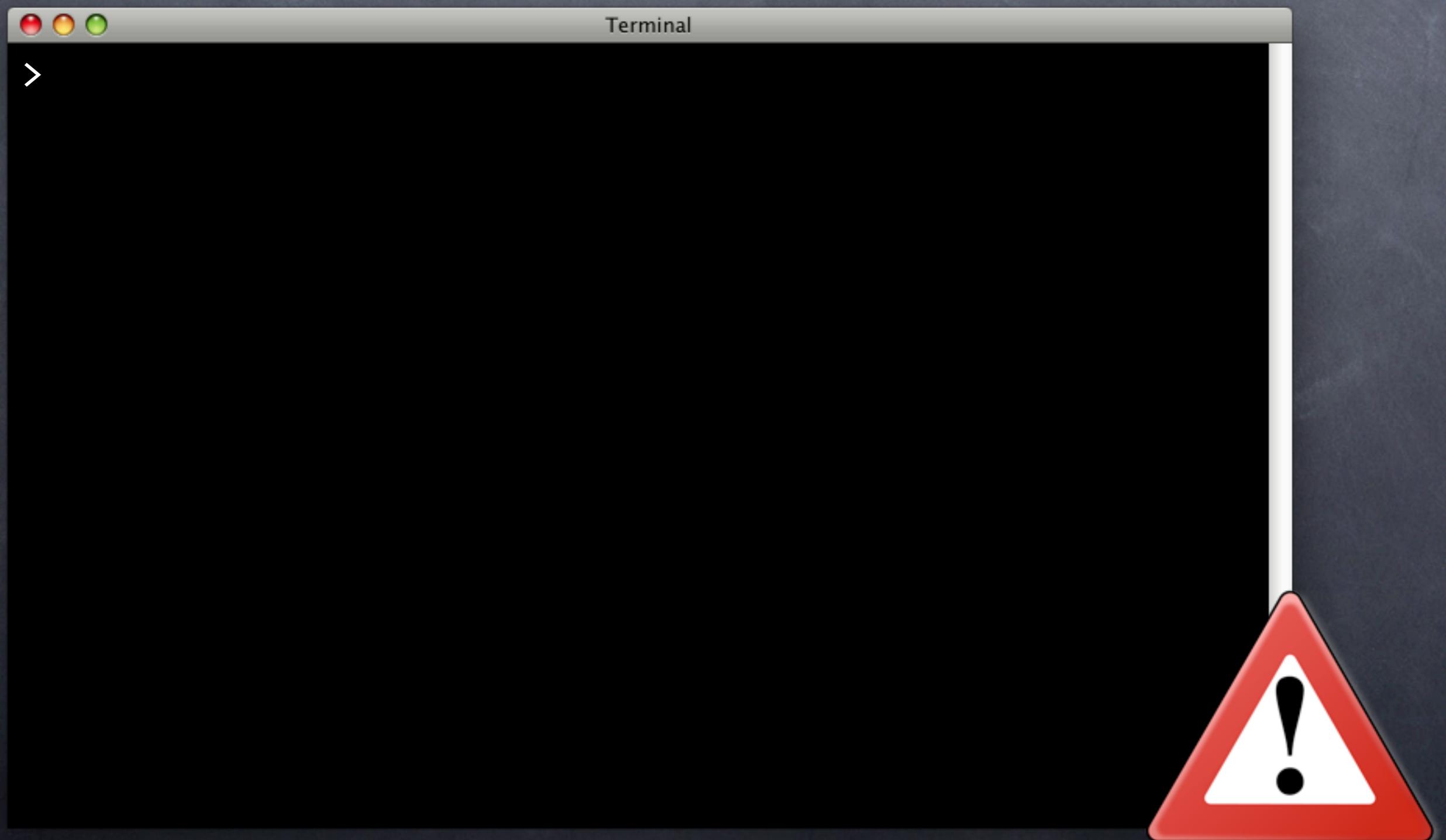
How's it Look?

```
Terminal  
> sqitch -d flipr_test status  
# On database flipr_test  
# Tag: HEAD+  
# Step: appuser  
# Date: 2012-05-18 17:15:23  
#  
Nothing to deploy (up-to-date)  
>
```



Proposed

Go Back



Go Back

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

A red triangular warning sign with a black exclamation mark in the center, positioned at the bottom right of the terminal window.

Go Back

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



Go Back

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

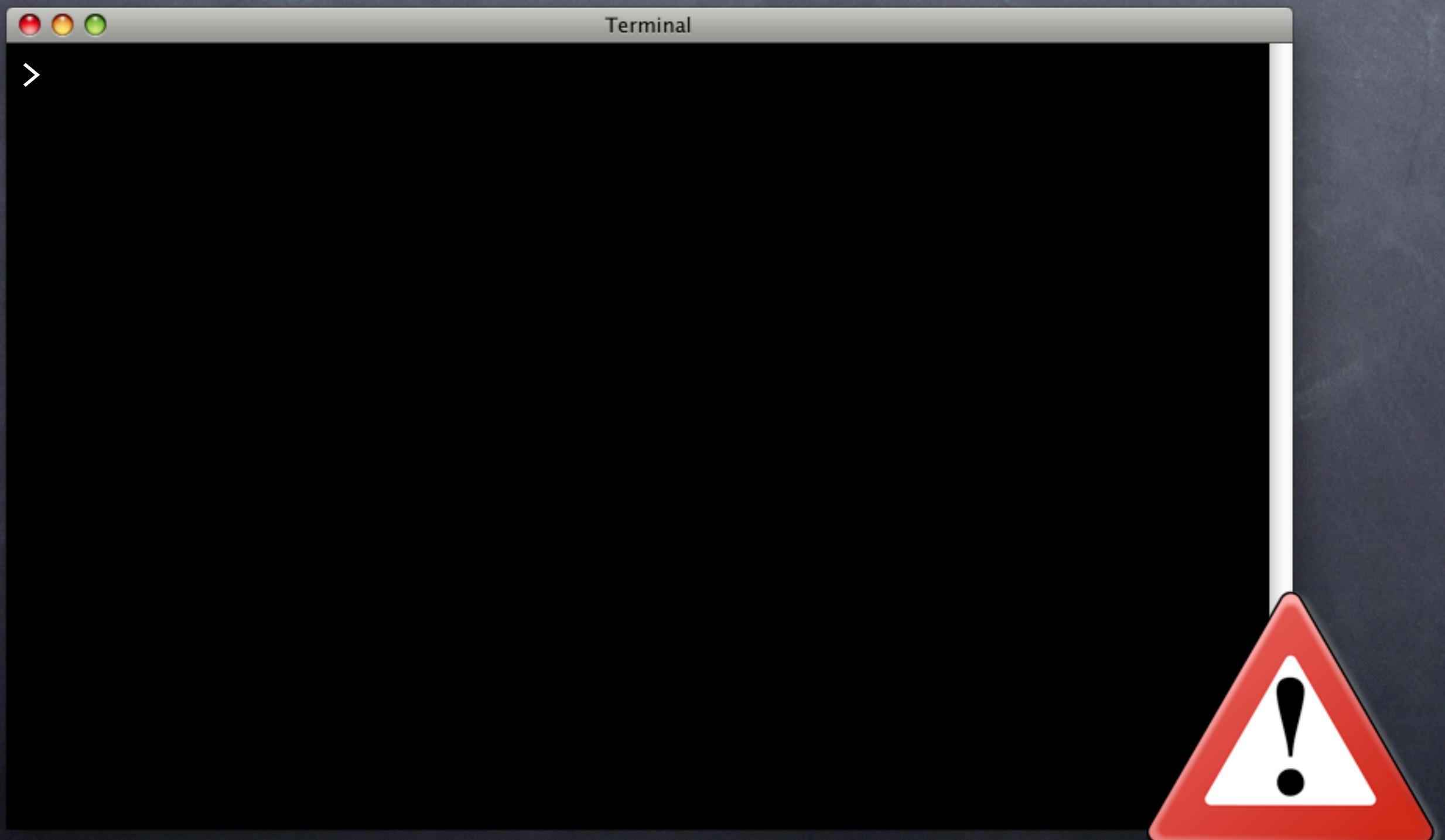
A red triangular warning sign with a black exclamation mark in the center, positioned to the right of the terminal window.

Go Back

```
Terminal  
> sqitch --db-name flipr_test revert  
Reverting all changes from flipr_test  
- appuser  
> psql -d flipr_test -c '\du flipr'  
      List of roles  
 Role name | Attributes | Member of  
-----+-----+-----  
>
```



What's The Status?



What's The Status?

```
Terminal  
> sqitch -d flipr_test status --untracked  
# On database flipr_test  
# Nothing deployed.  
#  
# Changes not deployed:  
# * HEAD+  
#   appuser  
#  
Use "sqitch deploy --untracked" to deploy these  
changes  
>
```

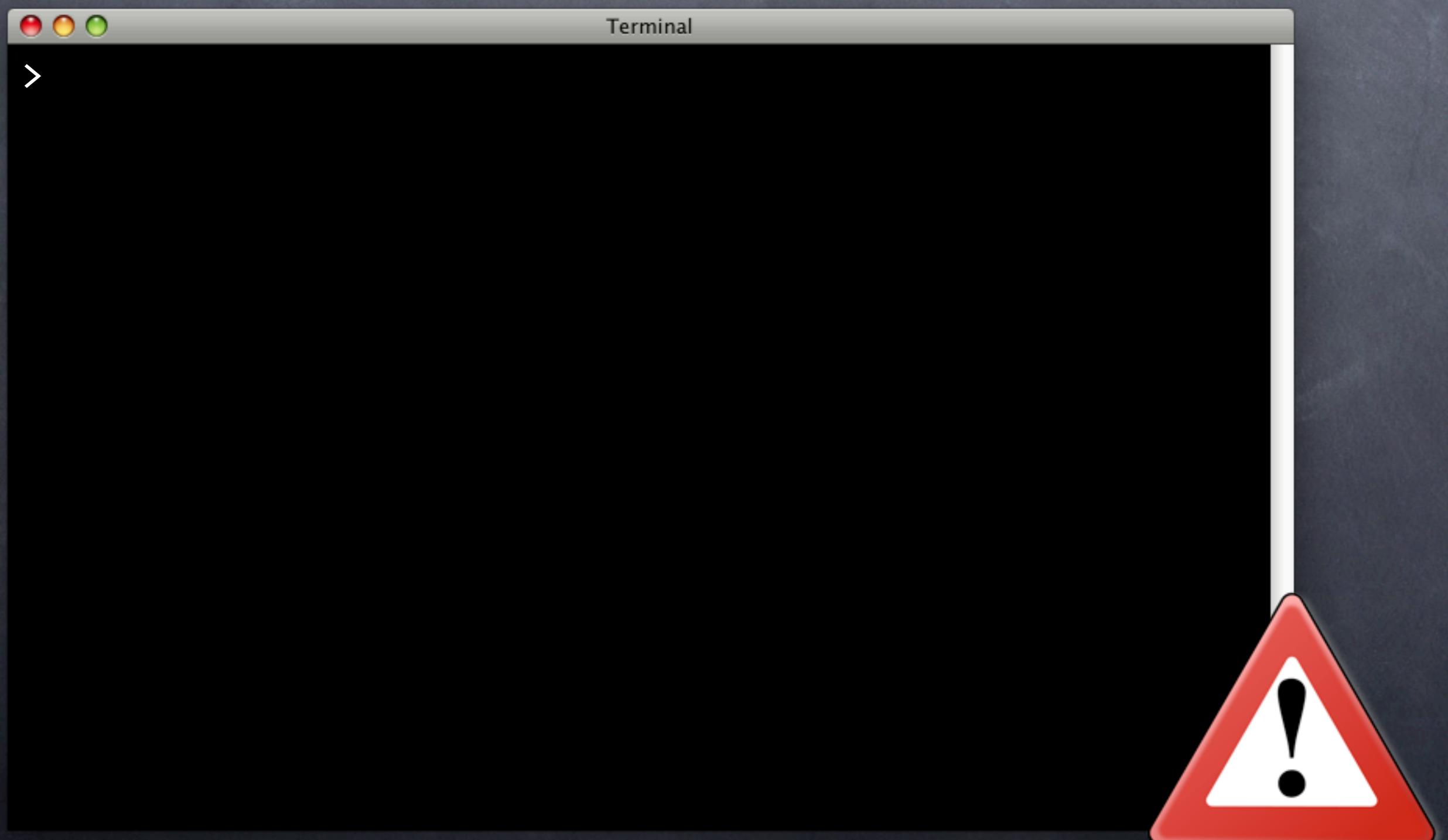


What's The Status?

```
Terminal  
> sqitch -d flipr_test status --untracked  
# On database flipr_test  
# Nothing deployed.  
  
#  
# Changes not deployed:  
# * HEAD+  
#   appuser  
  
#  
Use "sqitch deploy --untracked" to deploy these  
changes  
>
```



History



History

```
Terminal  
> sqitch -d flipr_test log  
step appuser reverted  
By: david  
Date: 2012-05-18 17:15:23  
Tag: HEAD+  
  
step appuser deployed  
By: david  
Date: 2012-05-18 17:18:12  
Tag: HEAD+  
>
```



History

```
Terminal  
> sqitch -d flipr_test log  
step appuser reverted  
By: david  
Date: 2012-05-18 17:15:23  
Tag: HEAD+  
  
step appuser deployed  
By: david  
Date: 2012-05-18 17:18:12  
Tag: HEAD+  
>
```

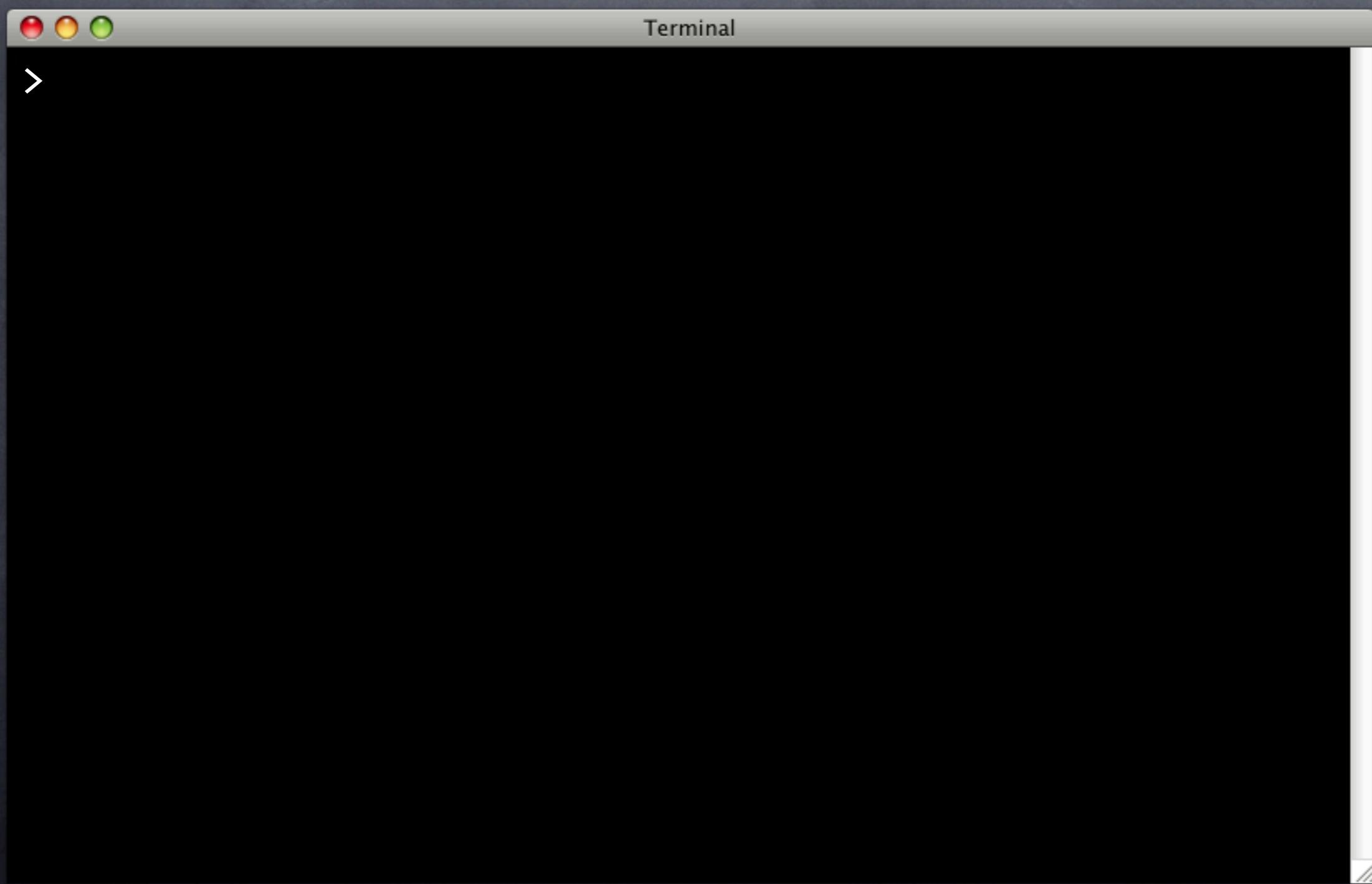


History

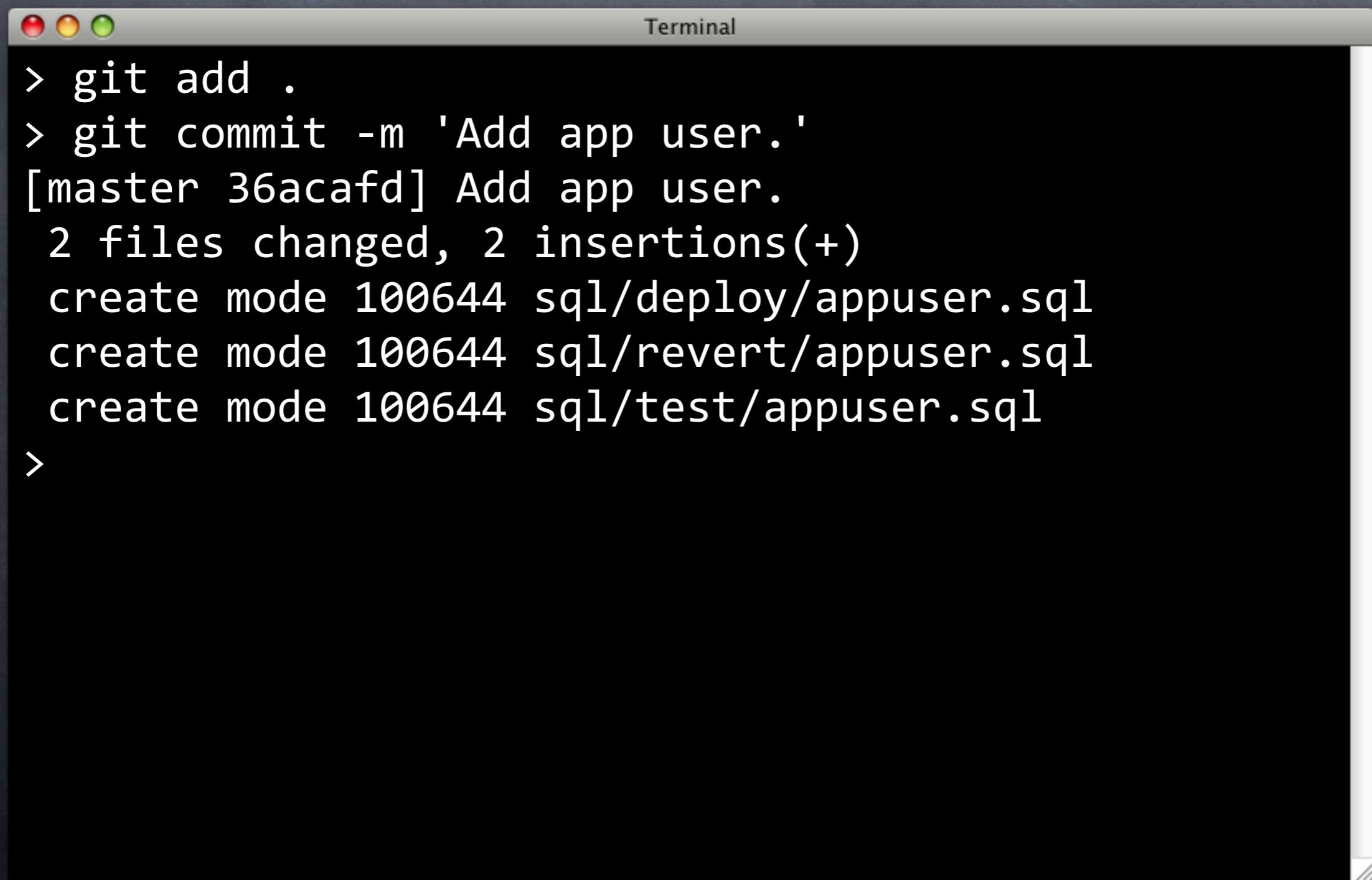
```
Terminal  
> sqitch -d flipr_test log  
step appuser reverted  
By: david  
Date: 2012-05-18 17:15:23  
Tag: HEAD+  
  
step appuser deployed  
By: david  
Date: 2012-05-18 17:18:12  
Tag: HEAD+  
>
```



Commit It!



Commit It!



```
Terminal  
> git add .  
> git commit -m 'Add app user.'  
[master 36acaf] Add app user.  
 2 files changed, 2 insertions(+)  
 create mode 100644 sql/deploy/appuser.sql  
 create mode 100644 sql/revert/appuser.sql  
 create mode 100644 sql/test/appuser.sql  
>
```

Commit It!

```
Terminal  
> git add .  
> git commit -m 'Add app user.'  
[master 36acaf] Add app user.  
 2 files changed, 2 insertions(+)  
 create mode 100644 sql/deploy/appuser.sql  
 create mode 100644 sql/revert/appuser.sql  
 create mode 100644 sql/test/appuser.sql  
> sqitch --db-name flipr_test deploy  
Deploying 36acaf to flipr_test  
  + appuser  
>
```



Commit It!

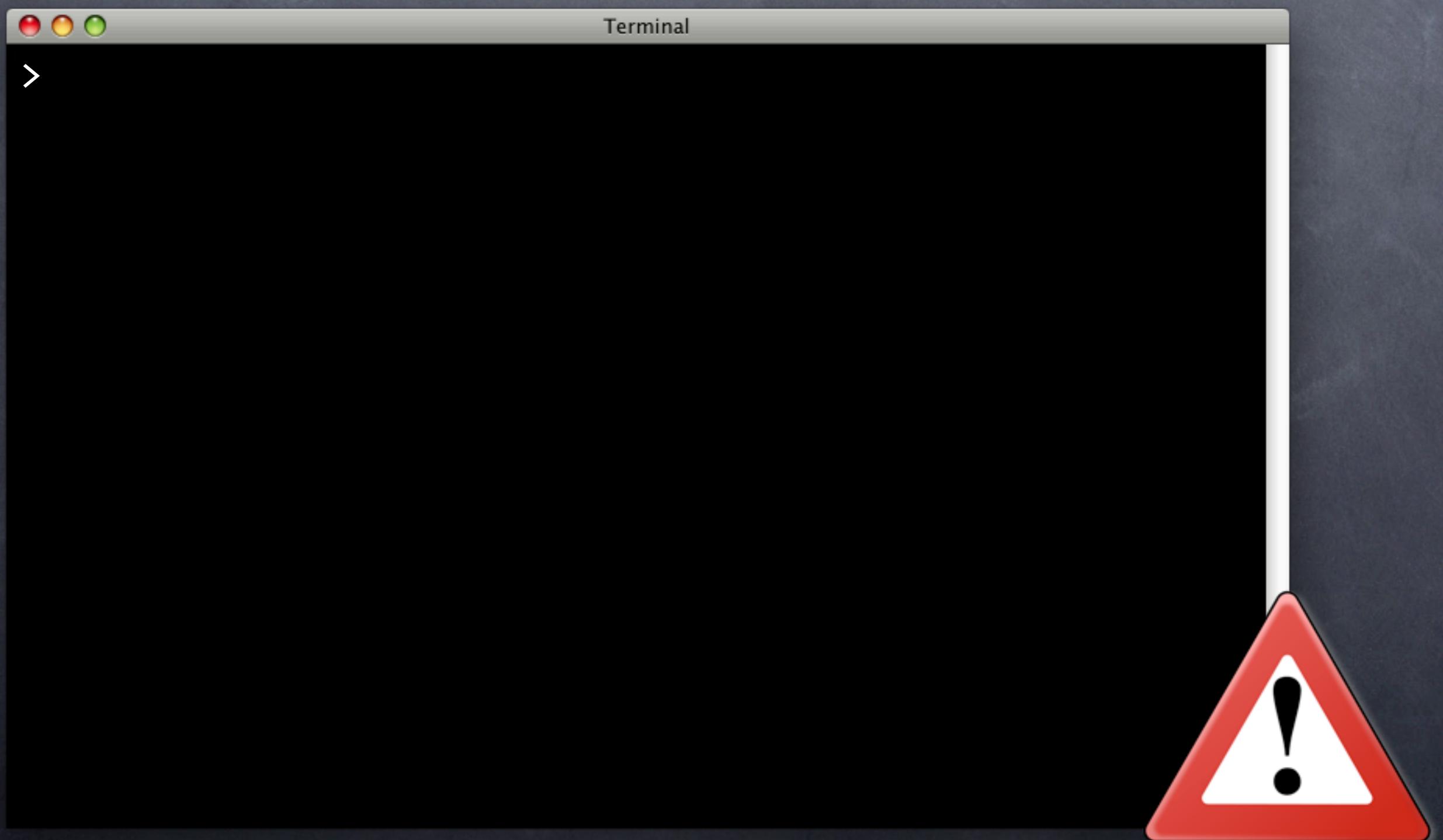
```
Terminal  
> git add .  
> git commit -m 'Add app user.'  
[master 36acaf] Add app user.  
 2 files changed, 2 insertions(+)  
 create mode 100644 sql/deploy/appuser.sql  
 create mode 100644 sql/revert/appuser.sql  
 create mode 100644 sql/test/appuser.sql  
> sqitch --db-name flipr_test deploy  
Deploying 36acaf to flipr_test  
  + appuser  
>
```



Commit It!

```
Terminal  
> git add .  
> git commit -m 'Add app user.'  
[master 36acaf] Add app user.  
 2 files changed, 2 insertions(+)  
 create mode 100644 sql/deploy/appuser.sql  
 create mode 100644 sql/revert/appuser.sql  
 create mode 100644 sql/test/appuser.sql  
> sqitch --db-name flipr_test deploy  
Deploying 36acaf to flipr_test  
  + appuser  
> psql -d flipr_test -c '\du flipr'  
      List of roles  
 Role name | Attributes | Member of  
-----+-----+-----  
 flipr | | {}
```

Status Update

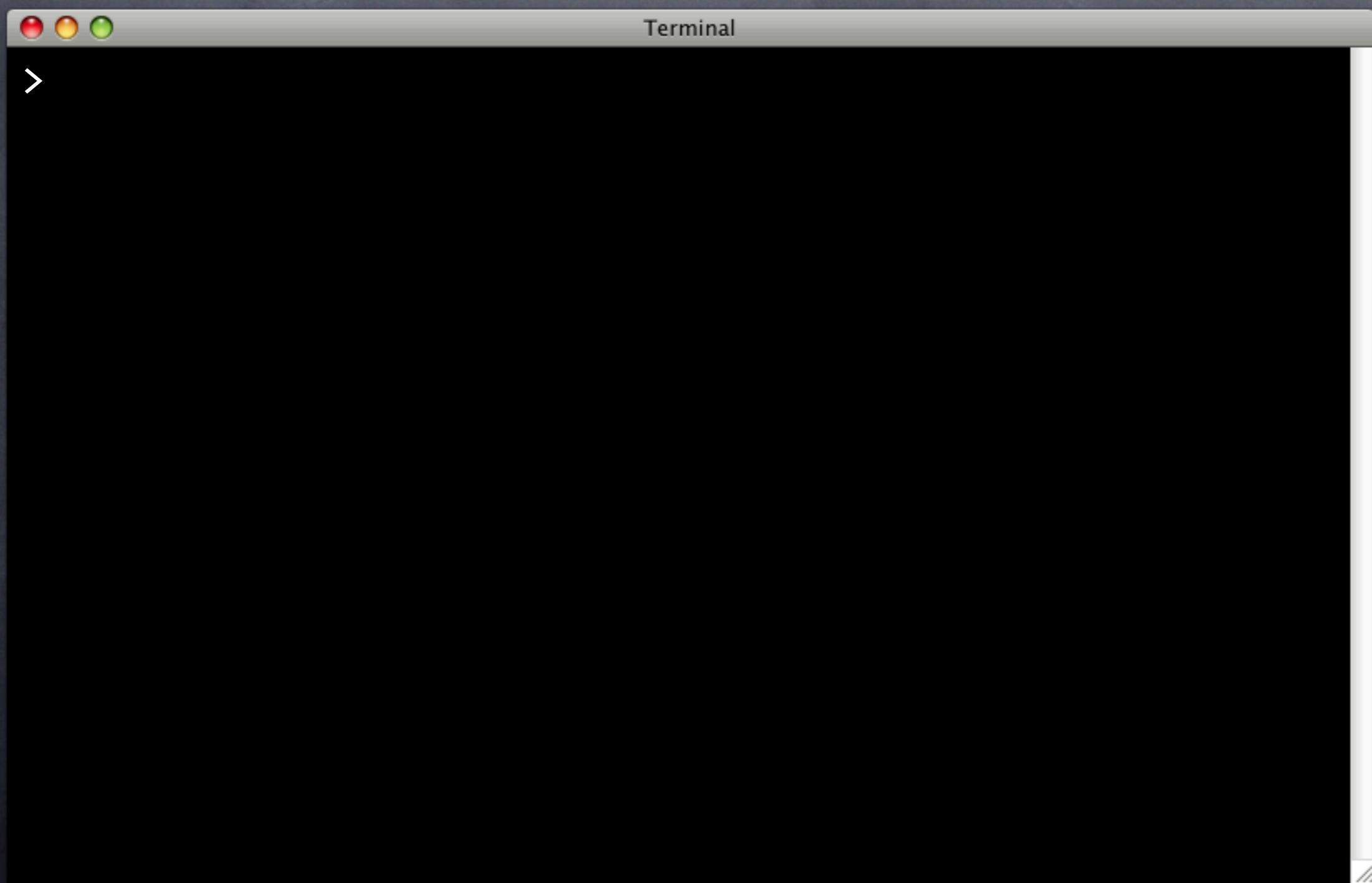


Status Update

```
Terminal  
> sqitch -d flipr_test status  
# On database flipr_test  
# Tag: 36acafdf  
# Step: appuser  
# Date: 2012-05-18 17:21:54  
#  
Nothing to deploy (up-to-date)  
>
```



Save My Fingers

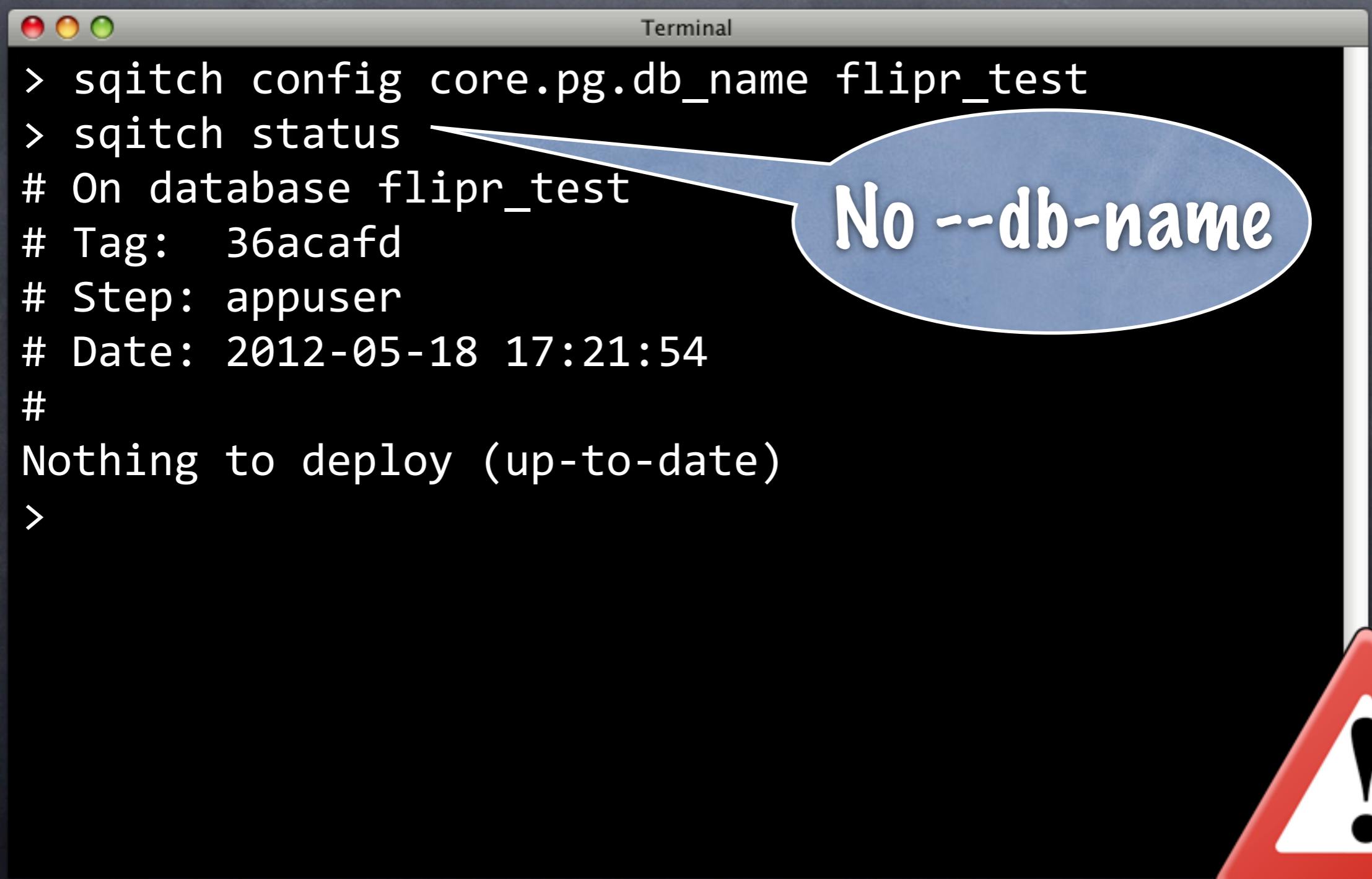


Save My Fingers



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

Save My Fingers



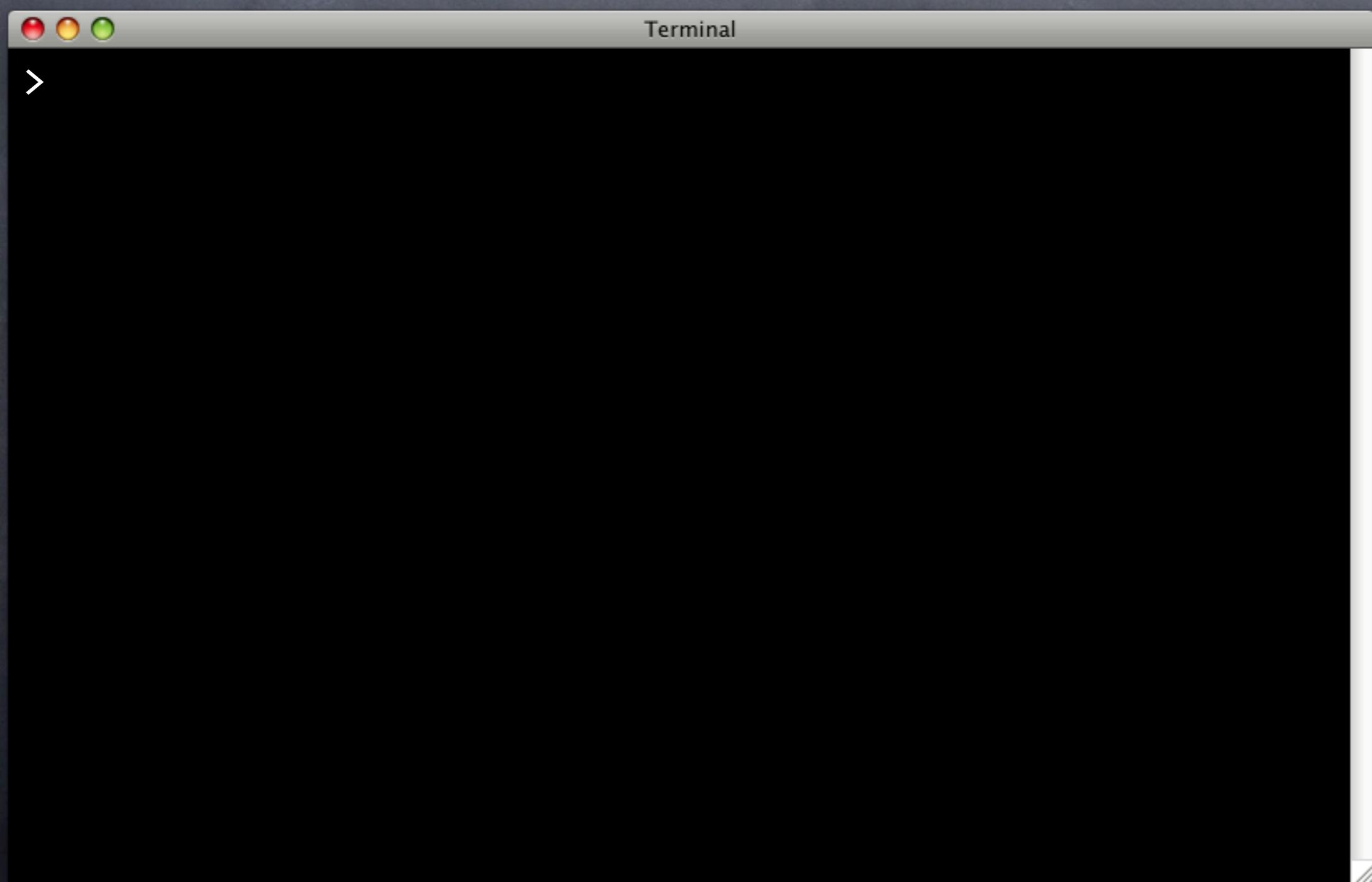
A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains the following text:

```
> sqitch config core.pg.db_name flipr_test
> sqitch status
# On database flipr_test
# Tag: 36acafdf
# Step: appuser
# Date: 2012-05-18 17:21:54
#
Nothing to deploy (up-to-date)
>
```

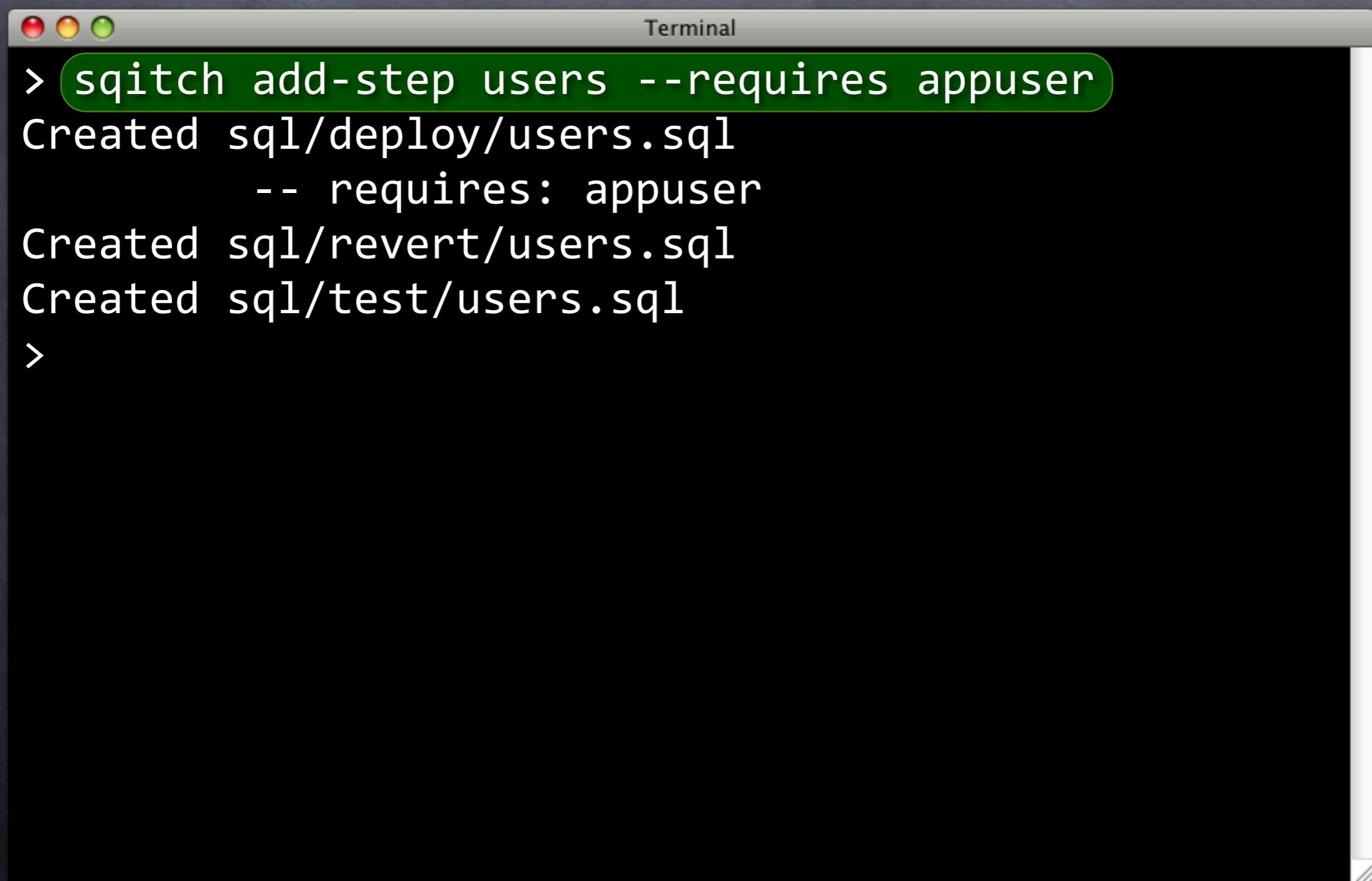
A blue callout bubble originates from the word "status" in the command history and points towards the right. Inside the bubble, the text "No --db-name" is displayed.



Dependencies!



Dependencies!

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-step users --requires appuser
Created sql/deploy/users.sql
    -- requires: appuser
Created sql/revert/users.sql
Created sql/test/users.sql
>
```

The first line of the command is highlighted with a green rounded rectangle.

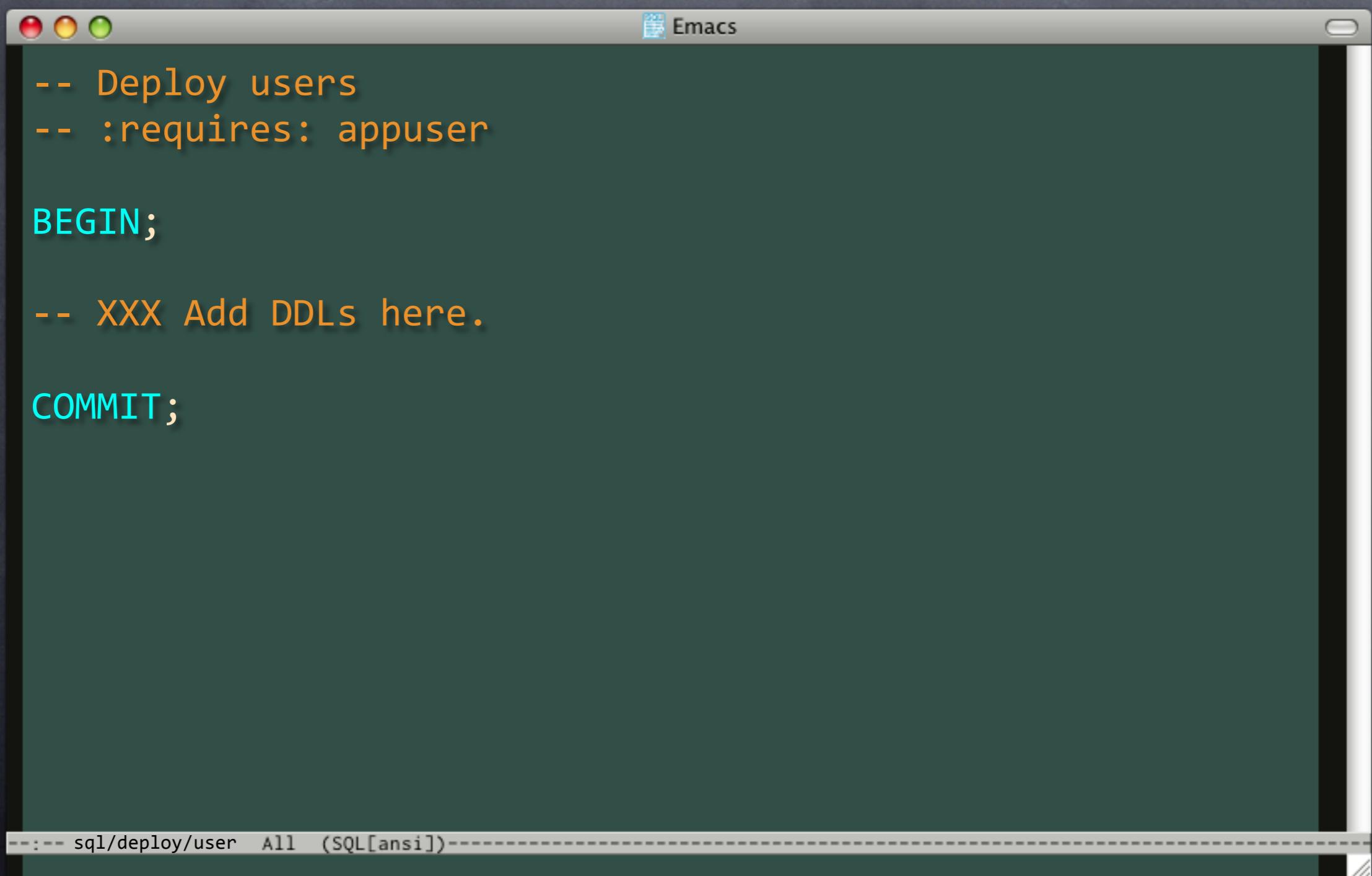
Dependencies!

```
Terminal  
> sqitch add-step users --requires appuser  
Created sql/deploy/users.sql  
    -- requires: appuser  
Created sql/revert/users.sql  
Created sql/test/users.sql  
>
```

Dependencies!

```
Terminal  
> sqitch add-step users --requires appuser  
Created sql/deploy/users.sql  
    -- requires: appuser  
Created sql/revert/users.sql  
Created sql/test/users.sql  
> emacs sql/deploy/users.sql
```

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 deployment script:

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

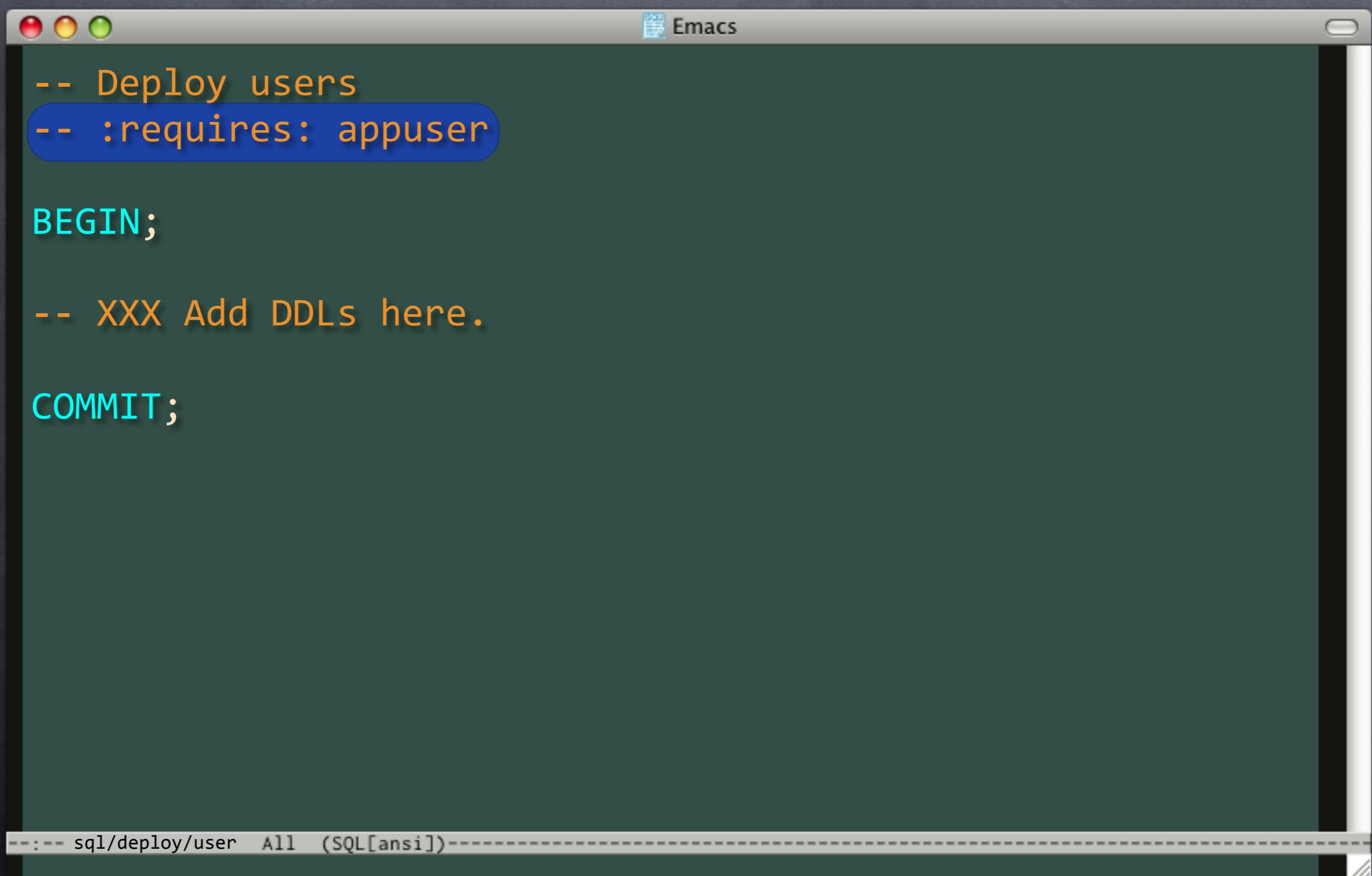
BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

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

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 script:

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

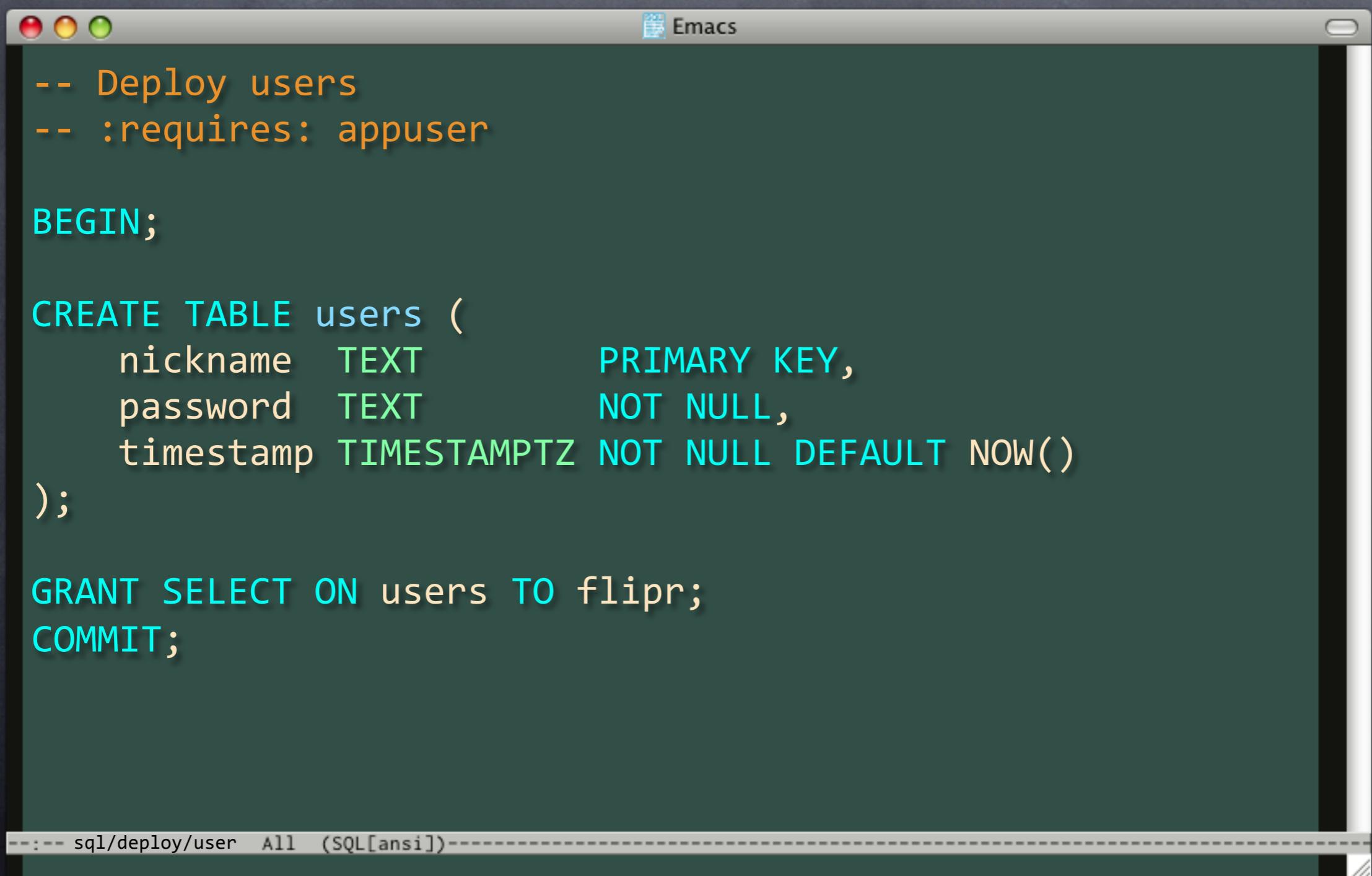
BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

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

sql/deploy/users.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 users
-- :requires: appuser

BEGIN;

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

GRANT SELECT ON users TO flipr;
COMMIT;

--- sql/deploy/user All (SQL[ansi])---
```

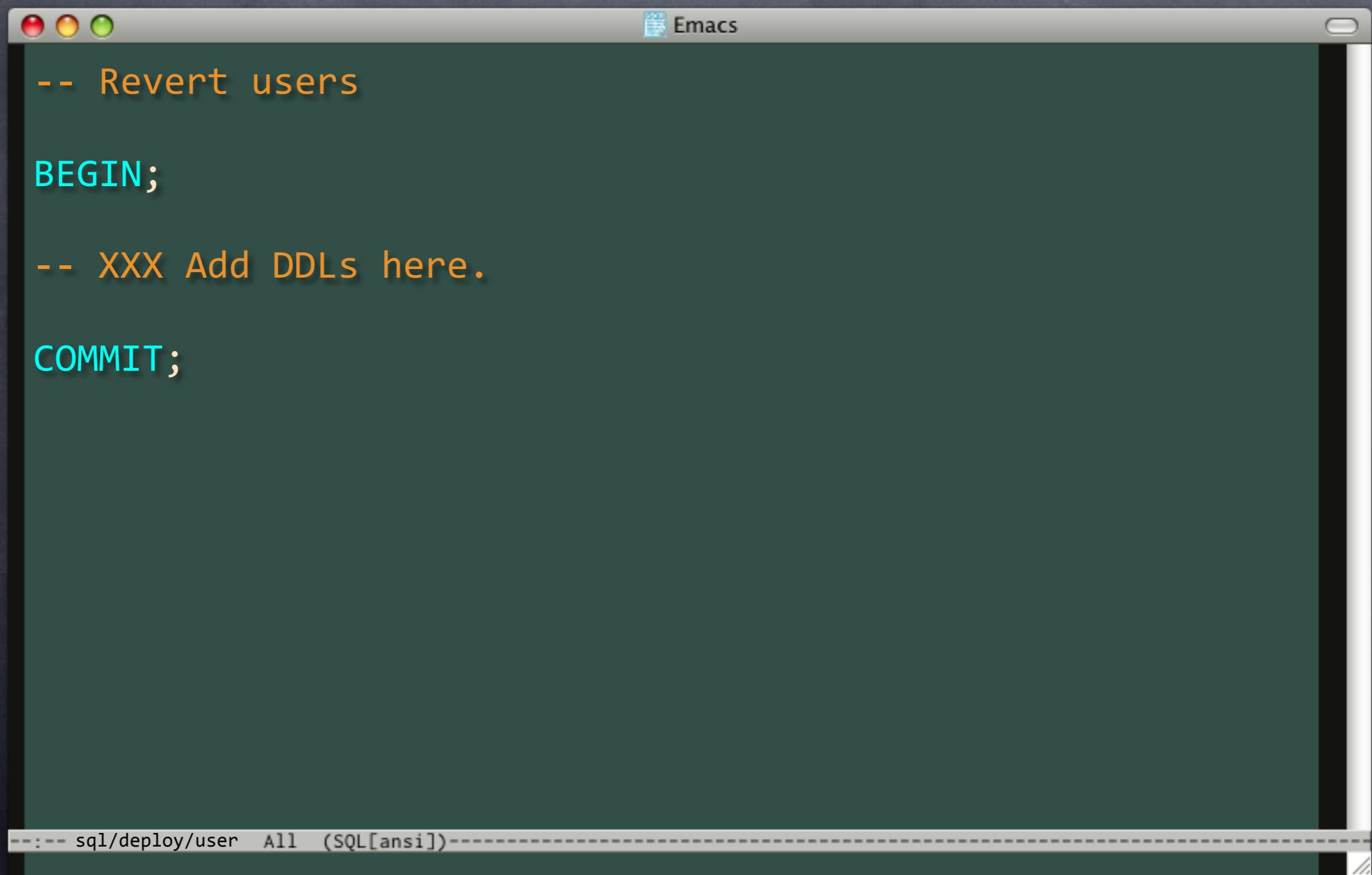
Dependencies!

```
Terminal  
> sqitch add-step users --requires appuser  
Created sql/deploy/users.sql  
    -- requires: appuser  
Created sql/revert/users.sql  
Created sql/test/users.sql  
> emacs sql/deploy/users.sql  
>
```

Dependencies!

```
Terminal  
> sqitch add-step users --requires appuser  
Created sql/deploy/users.sql  
    -- requires: appuser  
Created sql/revert/users.sql  
Created sql/test/users.sql  
> emacs sql/deploy/users.sql  
> emacs sql/revert/users.sql
```

sql/revert/users



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

```
-- Revert users

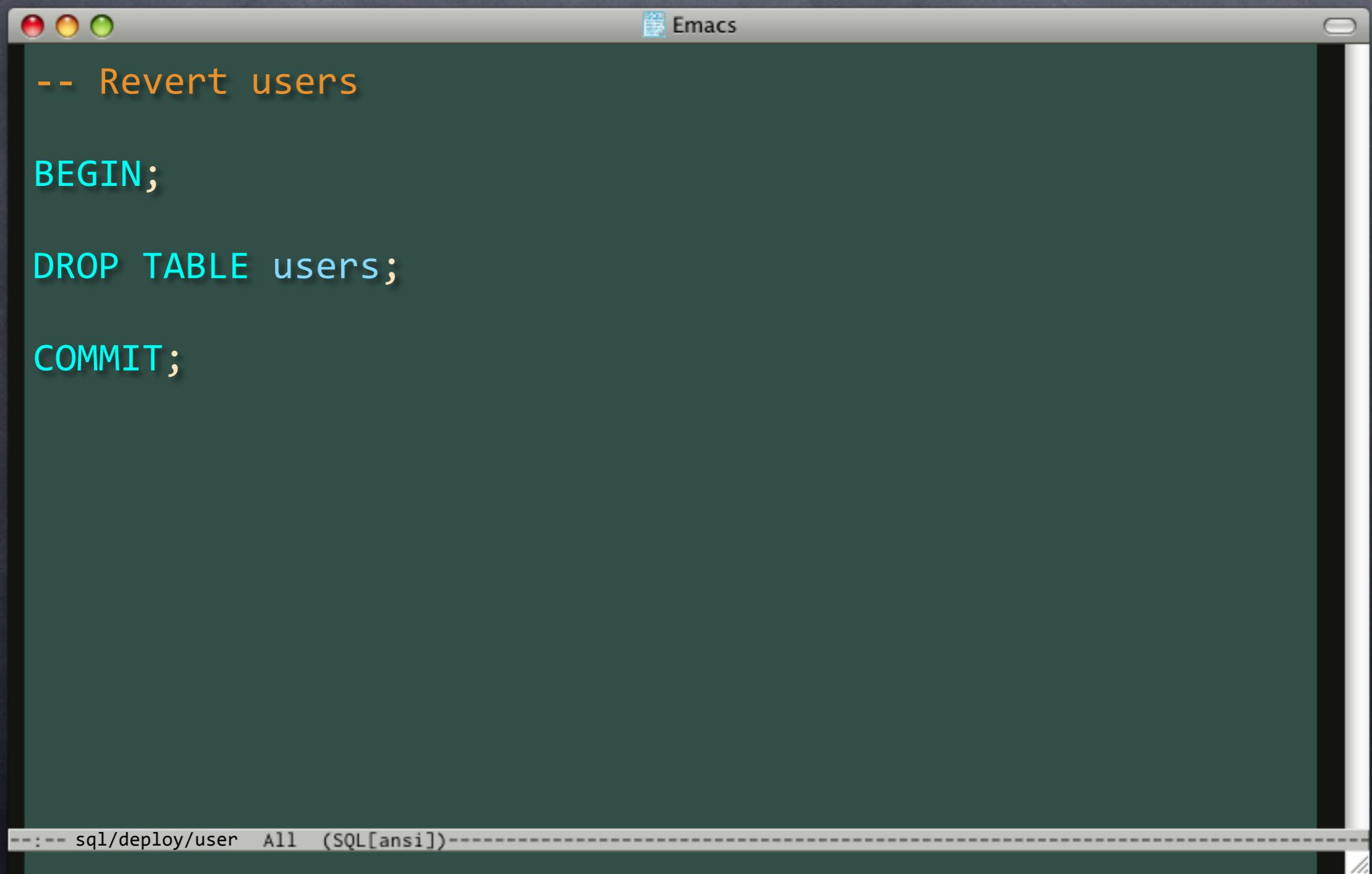
BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

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

sql/revert/users



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

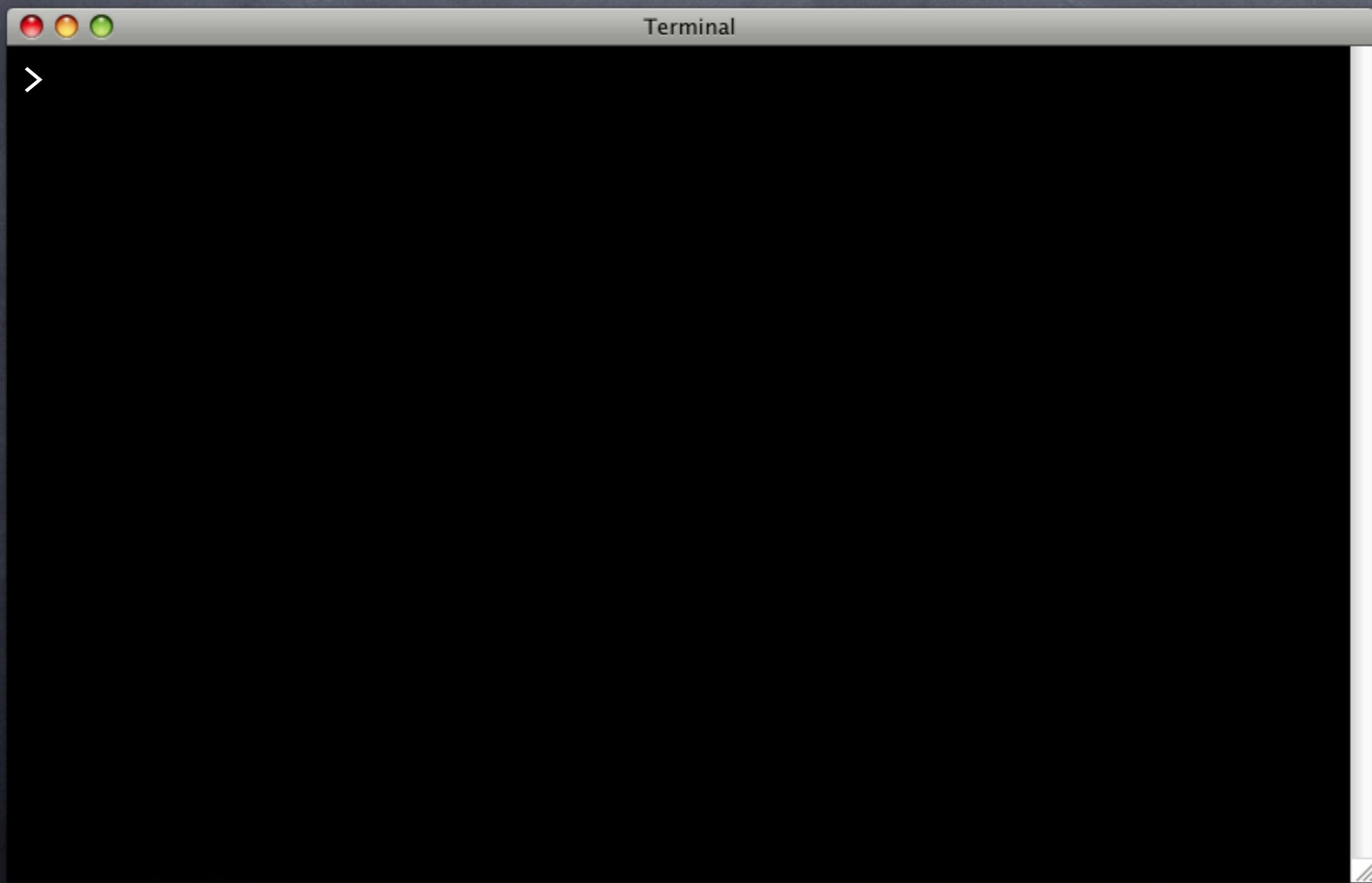
BEGIN;

DROP TABLE users;

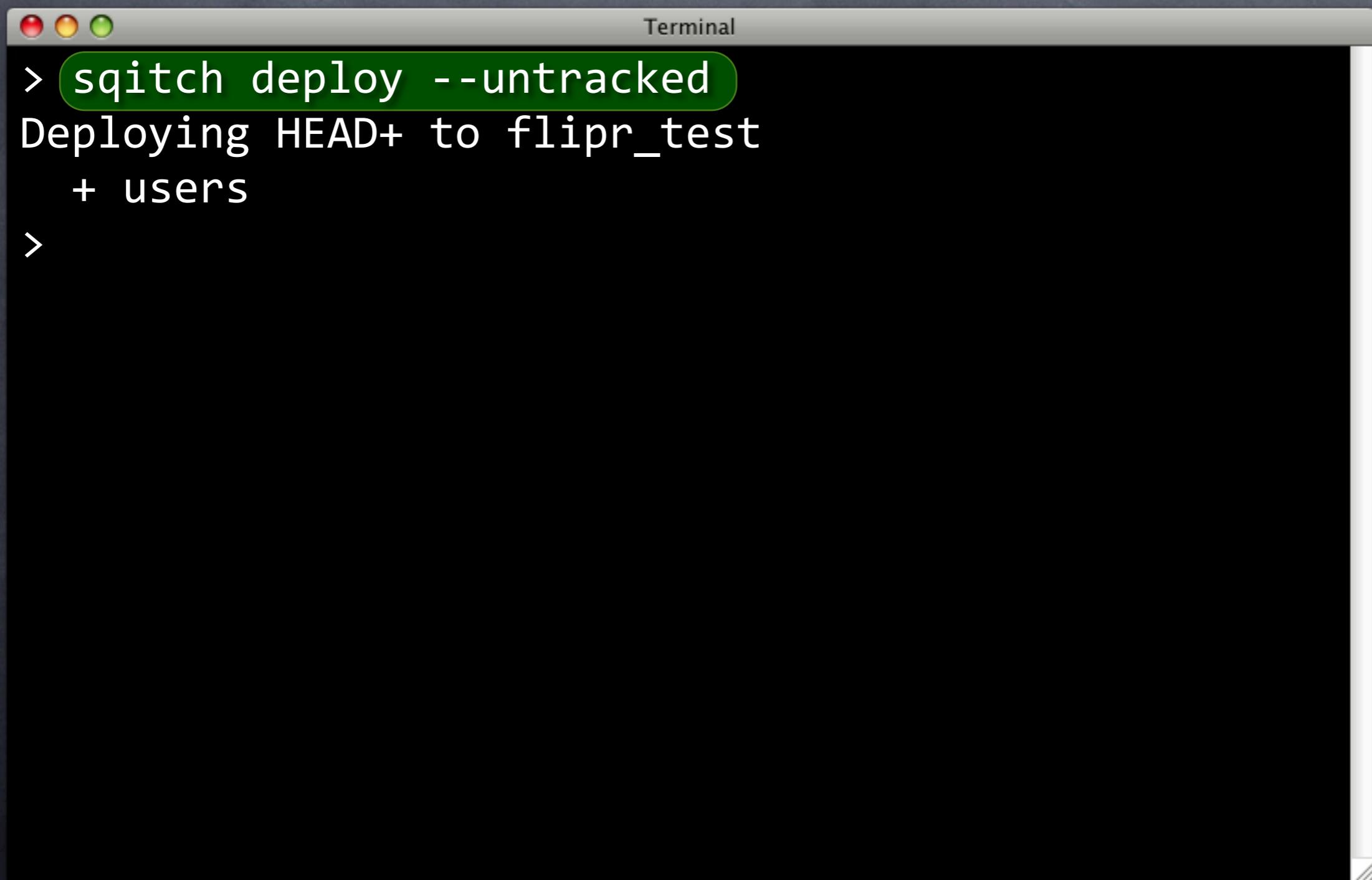
COMMIT;
```

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

Make Users

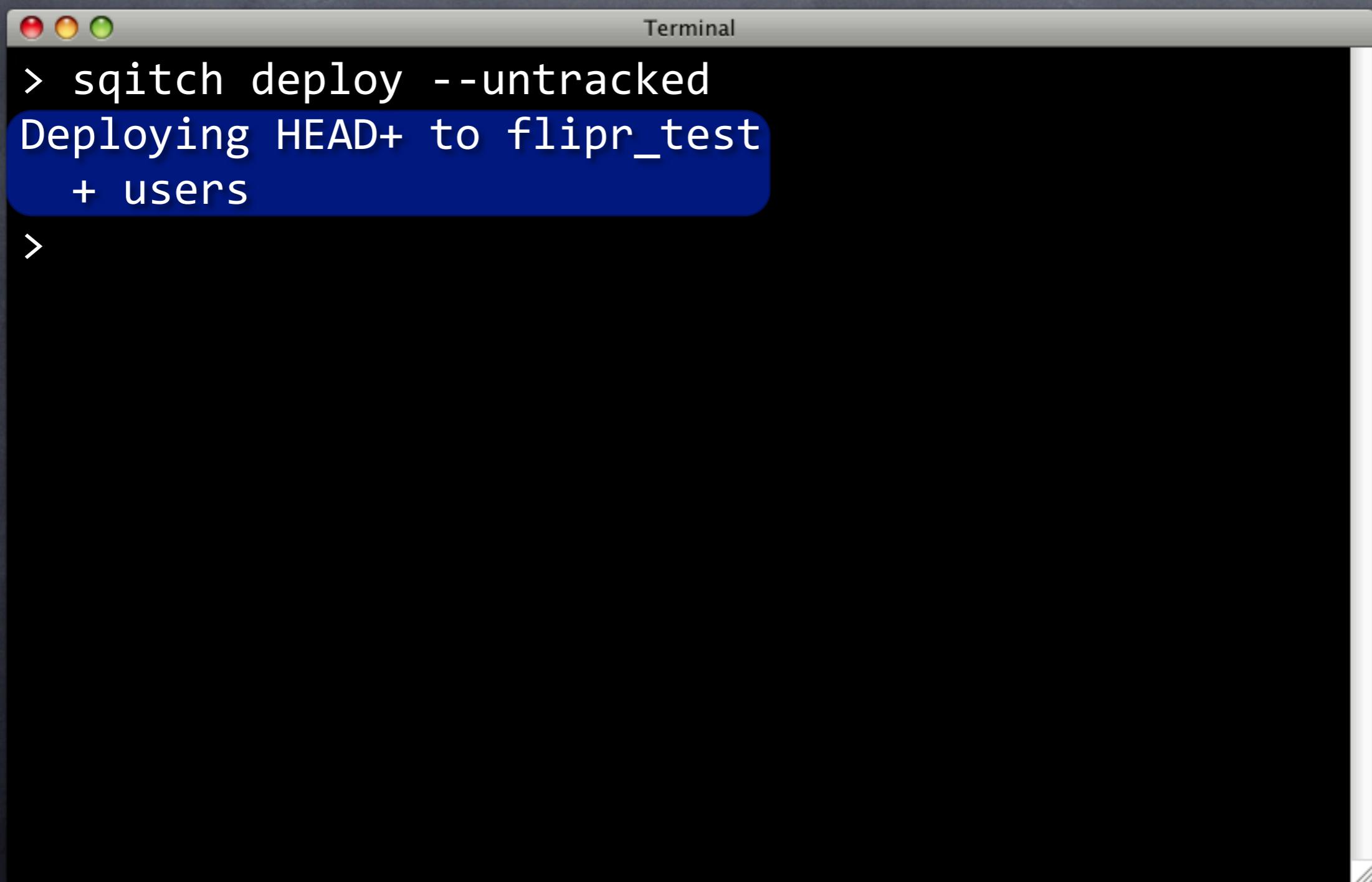


Make Users



```
Terminal  
> sqitch deploy --untracked  
Deploying HEAD+ to flipr_test  
+ users  
>
```

Make Users

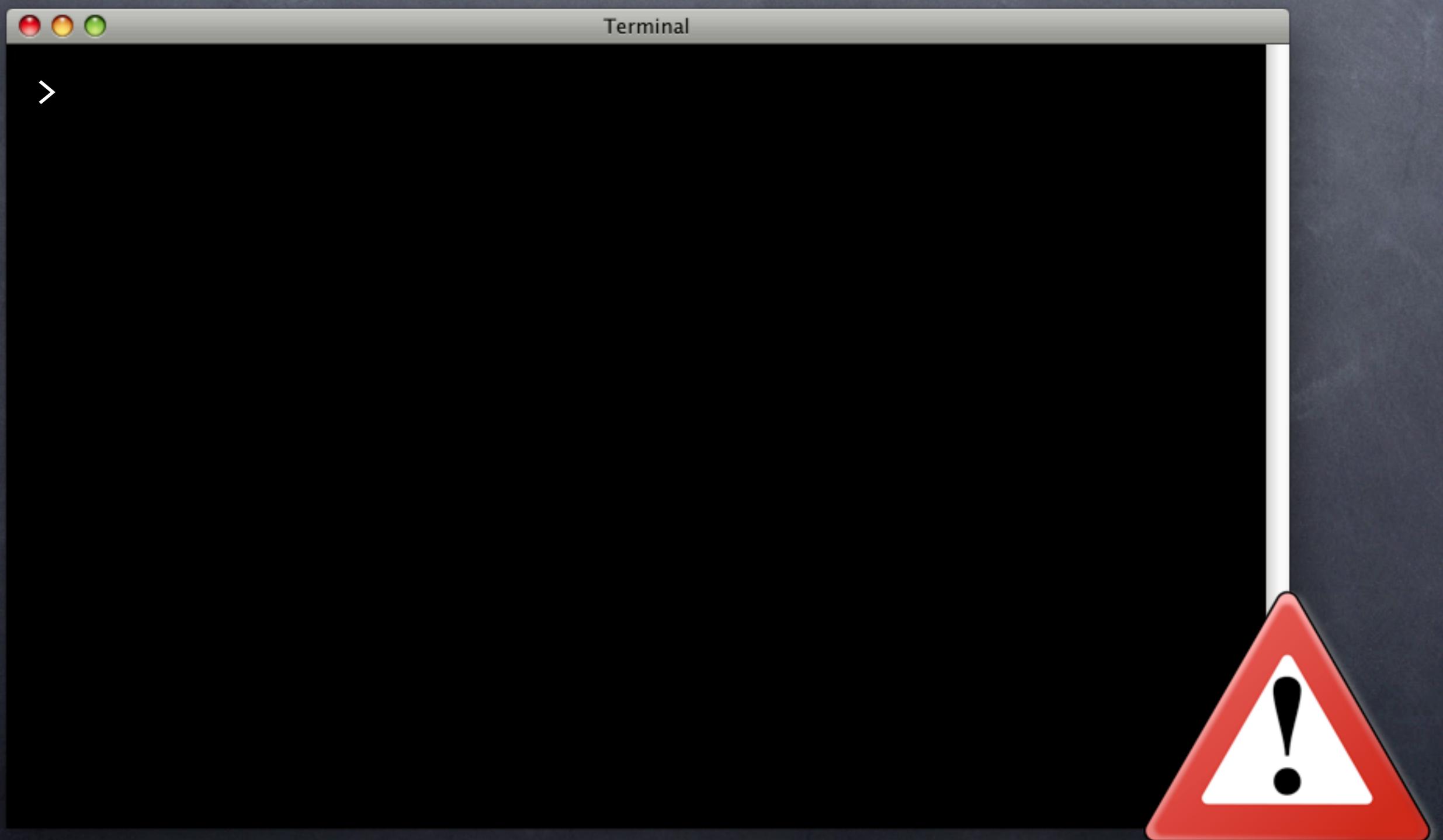


```
Terminal  
> sqitch deploy --untracked  
Deploying HEAD+ to flipr_test  
+ users  
>
```

Make Users

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

Status Update



Status Update

```
Terminal  
> sqitch status  
# On database flipr_test  
# Tag: HEAD+  
# Step: users  
# Date: 2012-05-18 17:25:39  
#  
Nothing to deploy (up-to-date)  
>
```



Status Update

```
Terminal  
> sqitch status  
# On database flipr_test  
# Tag: HEAD+  
# Step: users  
# Date: 2012-05-18 17:25:39  
#  
Nothing to deploy (up-to-date)  
> sqitch revert --to HEAD  
Reverting HEAD+ from flipr_test  
- users  
>
```



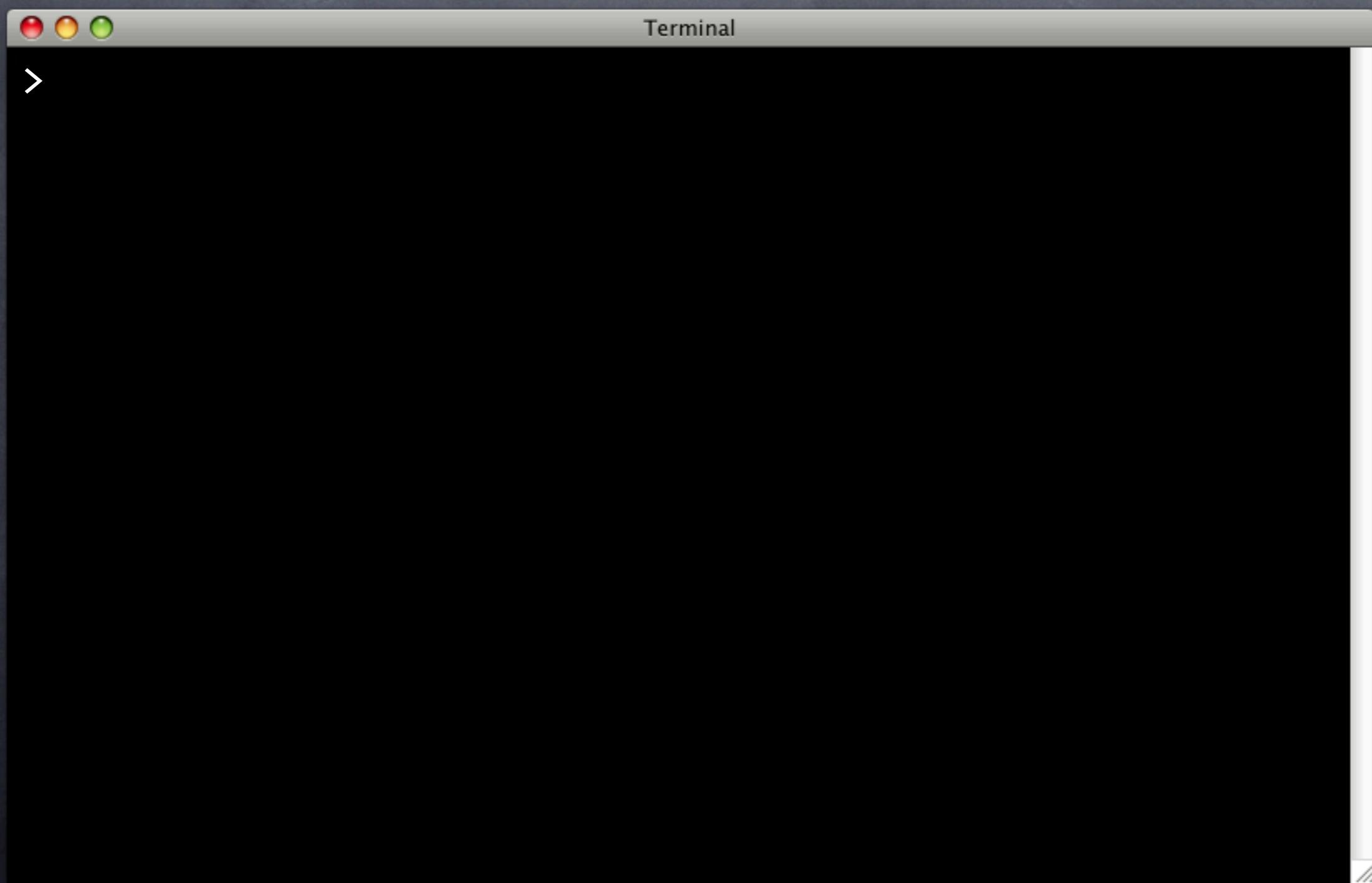
Status Update

```
Terminal

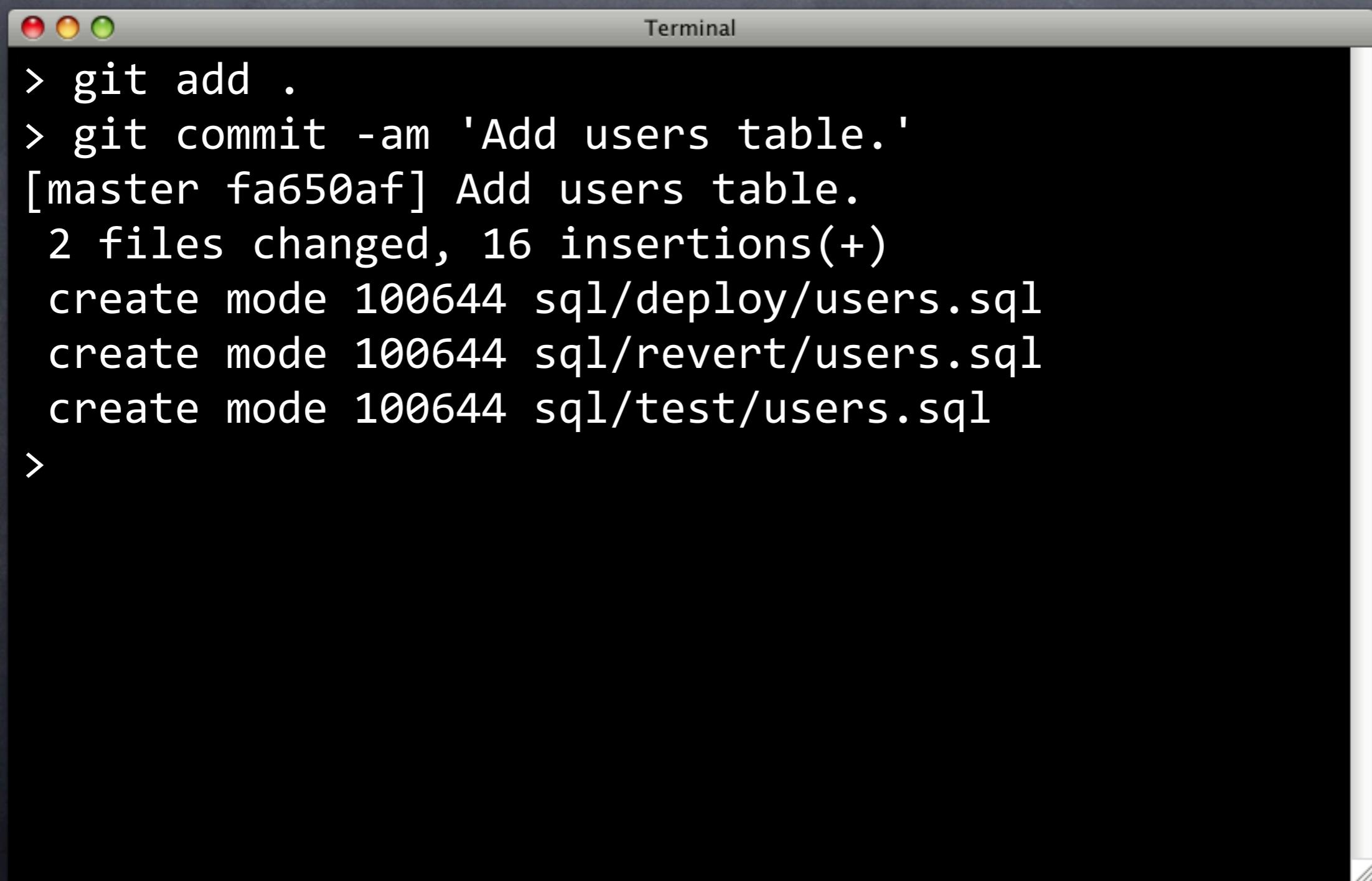
> sqitch status
# On database flipr_test
# Tag: HEAD+
# Step: users
# Date: 2012-05-18 17:25:39
#
Nothing to deploy (up-to-date)
> sqitch revert --to HEAD
Reverting HEAD+ from flipr_test
- users
>
```



Check in and Deploy



Check in and Deploy



A screenshot of a Mac OS X Terminal window titled "Terminal". The window contains a command-line session for version control:

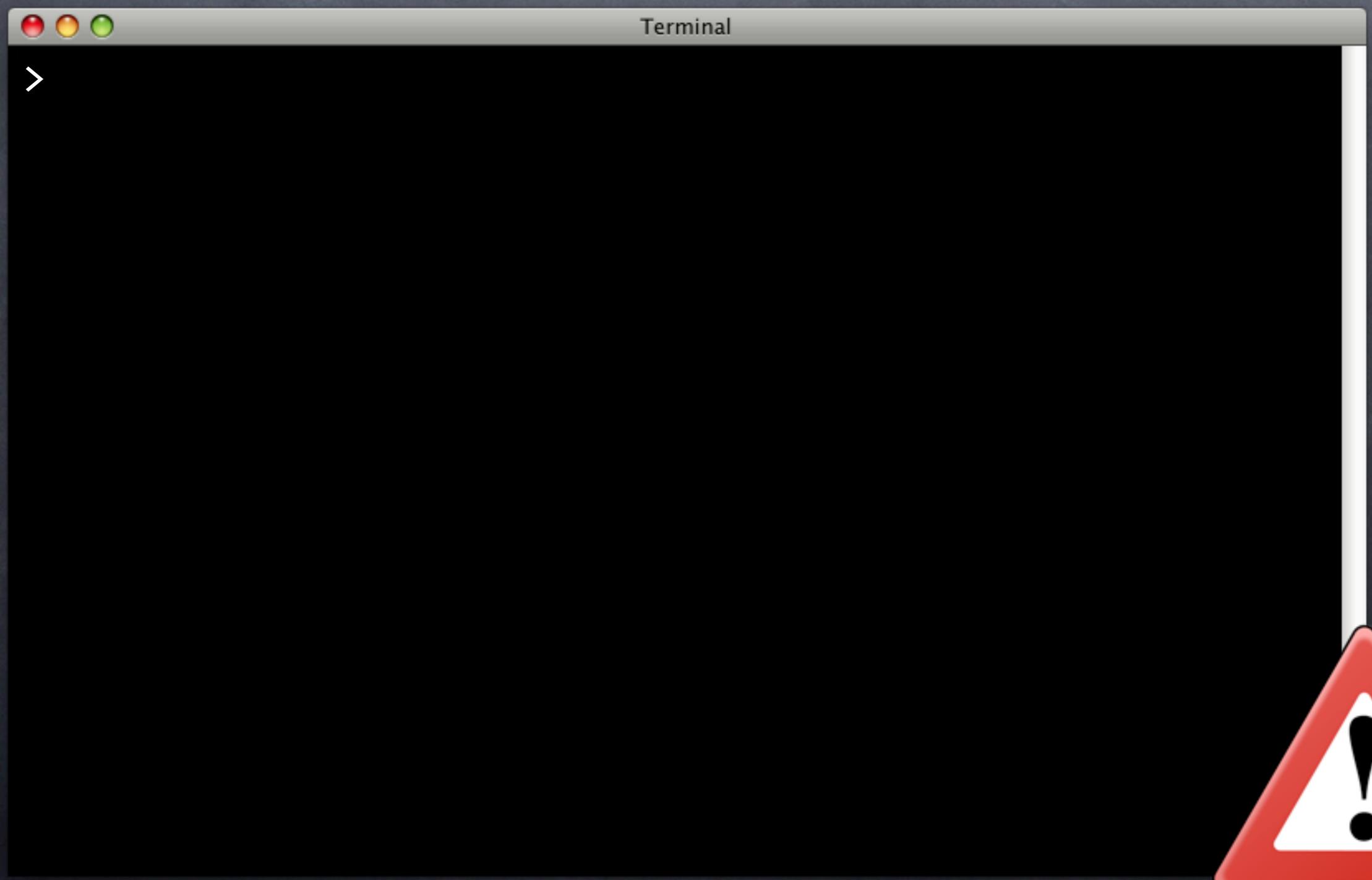
```
> git add .
> git commit -am 'Add users table.'
[master fa650af] Add users table.
 2 files changed, 16 insertions(+)
 create mode 100644 sql/deploy/users.sql
 create mode 100644 sql/revert/users.sql
 create mode 100644 sql/test/users.sql
>
```

Check in and Deploy



```
Terminal
> git add .
> git commit -am 'Add users table.'
[master fa650af] Add users table.
 2 files changed, 16 insertions(+)
 create mode 100644 sql/deploy/users.sql
 create mode 100644 sql/revert/users.sql
 create mode 100644 sql/test/users.sql
> sqitch deploy
Deploying fa650af to flipr_test
  + users
>
```

Up to Date



Up to Date

```
Terminal  
> sqitch status --show tags  
# On database flipr_test  
# Tag: fa650af  
# Step: users  
# Date: 2012-05-18 17:31:23  
#  
# Tags:  
#     fa650af - 2012-05-18 17:31:23 - david  
#     36acafdf - 2012-05-18 17:25:39 - david  
#  
Nothing to deploy (up-to-date)  
>
```



Up to Date

```
Terminal  
> sqitch status --show tags  
# On database flipr_test  
# Tag: fa650af  
# Step: users  
# Date: 2012-05-18 17:31:23  
#  
# Tags:  
#     fa650af - 2012-05-18 17:31:23 - david  
#     36acafdf - 2012-05-18 17:25:39 - david  
#  
Nothing to deploy (up-to-date)  
>
```

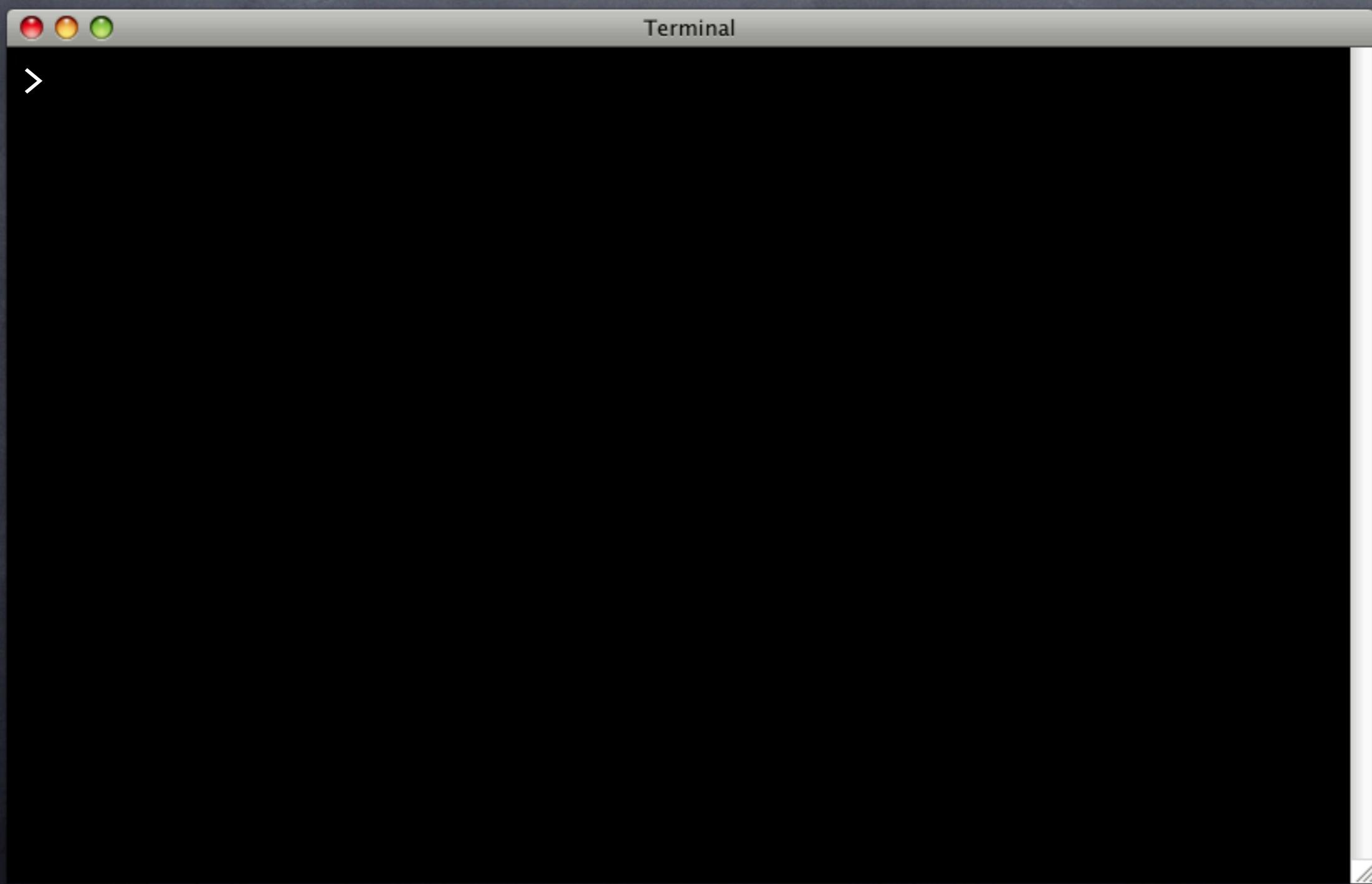


Up to Date

```
Terminal  
> sqitch status --show tags  
# On database flipr_test  
# Tag: fa650af  
# Step: users  
# Date: 2012-05-18 17:31:23  
#  
# Tags:  
#     fa650af - 2012-05-18 17:31:23 - david  
#     36acafdf - 2012-05-18 17:25:39 - david  
#  
Nothing to deploy (up-to-date)  
>
```



A Twofer



A Twofer

```
Terminal  
> sqitch add-step insert_user -r users -r appuser  
Adding sql/deploy/insert_user.sql  
    -- requires: users, appuser  
Adding sql/revert/insert_user.sql  
Adding sql/test/insert_user.sql  
  
>
```

A Twofer

```
Terminal  
> sqitch add-step insert_user -r users -r appuser  
Adding sql/deploy/insert_user.sql  
    -- requires: users, appuser  
Adding sql/revert/insert_user.sql  
Adding sql/test/insert_user.sql  
  
>
```

A Twofer

```
Terminal  
> sqitch add-step insert_user -r users -r appuser  
Adding sql/deploy/insert_user.sql  
-- requires: users, appuser  
Adding sql/revert/insert_user.sql  
Adding sql/test/insert_user.sql  
  
> sqitch add-step change_pass -r users -r appuser  
Adding sql/deploy/change_pass.sql  
-- requires: users, appuser  
Adding sql/revert/change_pass.sql  
Adding sql/test/change_pass.sql  
  
>
```

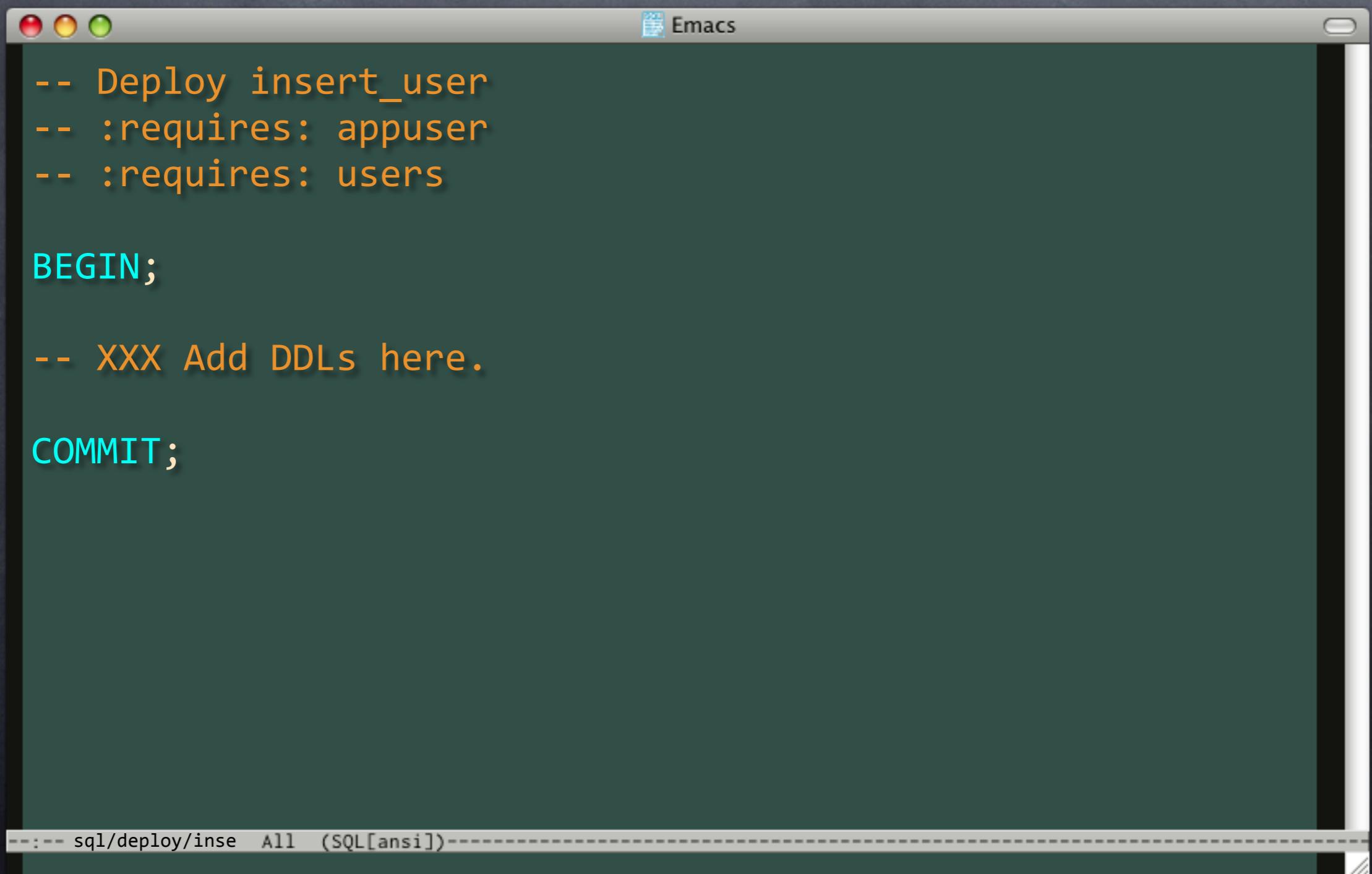
A Twofer

```
Terminal  
> sqitch add-step insert_user -r users -r appuser  
Adding sql/deploy/insert_user.sql  
  -- requires: users, appuser  
Adding sql/revert/insert_user.sql  
Adding sql/test/insert_user.sql  
  
> sqitch add-step change_pass -r users -r appuser  
Adding sql/deploy/change_pass.sql  
  -- requires: users, appuser  
Adding sql/revert/change_pass.sql  
Adding sql/test/change_pass.sql  
  
>
```

A Twofer

```
Terminal  
> sqitch add-step insert_user -r users -r appuser  
Adding sql/deploy/insert_user.sql  
  -- requires: users, appuser  
Adding sql/revert/insert_user.sql  
Adding sql/test/insert_user.sql  
  
> sqitch add-step change_pass -r users -r appuser  
Adding sql/deploy/change_pass.sql  
  -- requires: users, appuser  
Adding sql/revert/change_pass.sql  
Adding sql/test/change_pass.sql  
  
> emacs sql/deploy/insert_user.sql
```

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

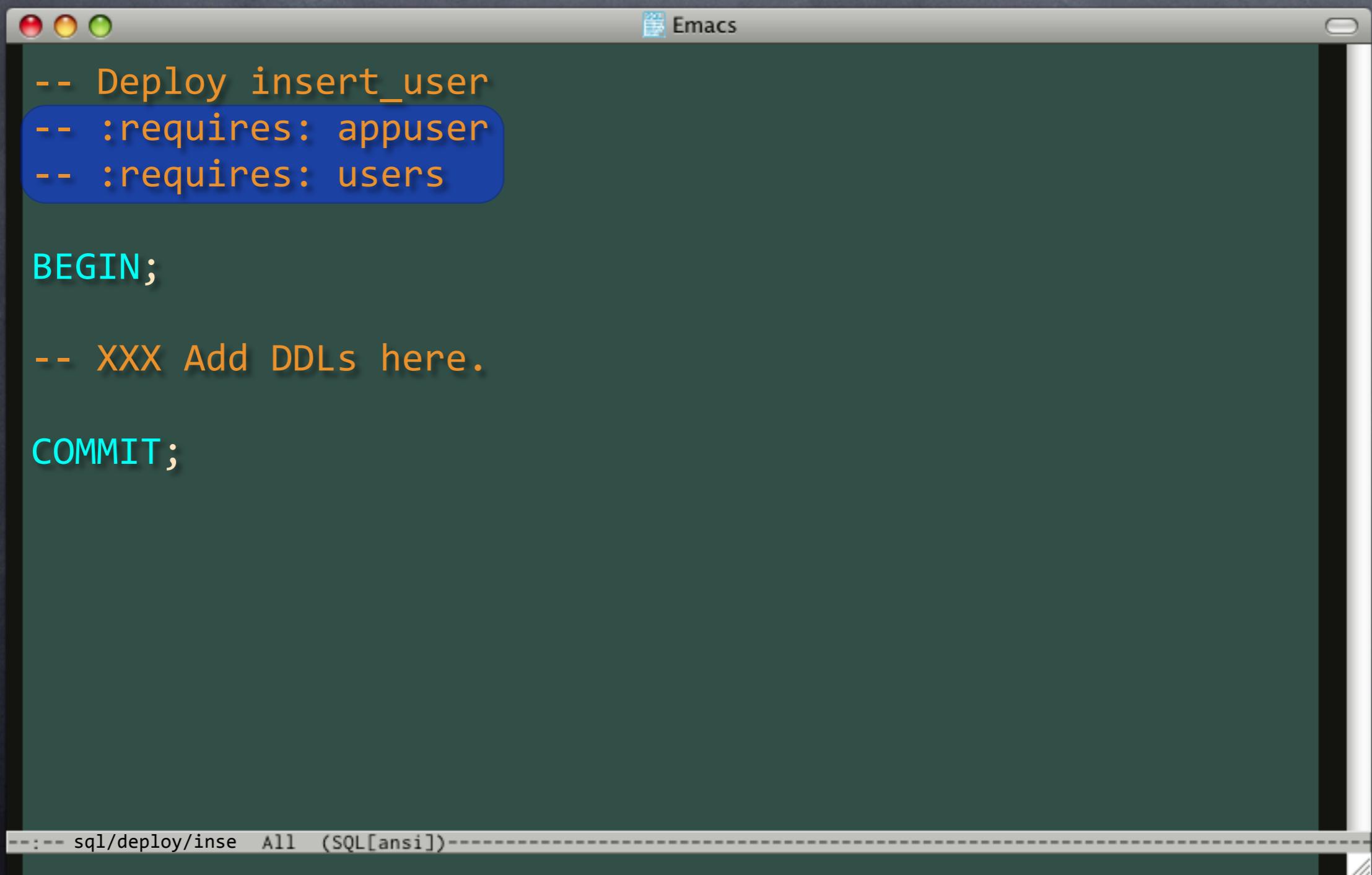
BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

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

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 "---- sql/deploy/inse All (SQL[ansi])----".

sql/deploy/ insert_user.sql

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

BEGIN;

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

GRANT EXECUTE ON FUNCTION insert_user(TEXT, TEXT) TO
flipr;
COMMIT;

--- sql/deploy/inse All (SQL[ansi])---
```

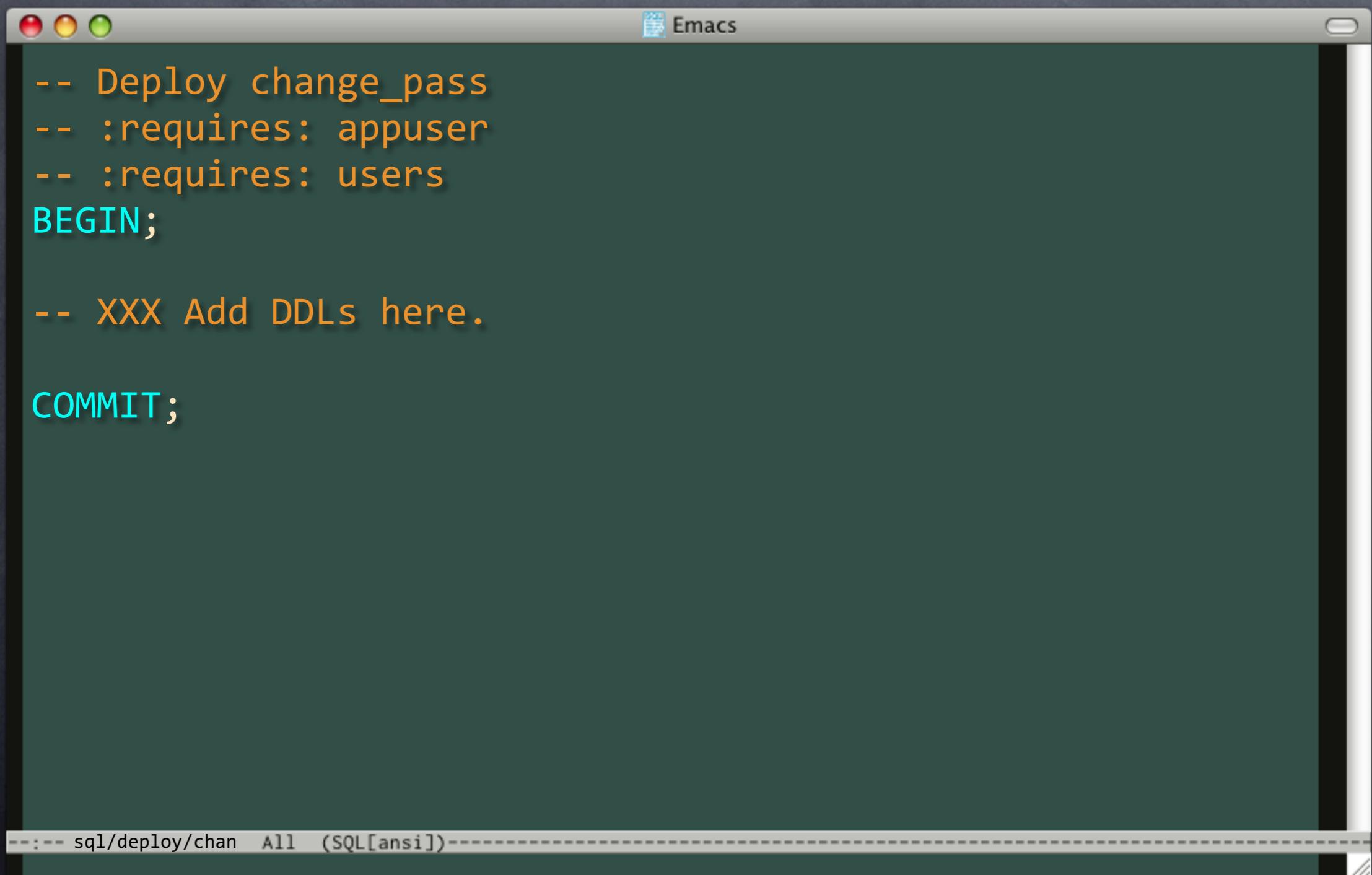
A Twofer

```
Terminal  
> sqitch add-step insert_user -r users -r appuser  
Adding sql/deploy/insert_user.sql  
-- requires: users, appuser  
Adding sql/revert/insert_user.sql  
Adding sql/test/insert_user.sql  
  
> sqitch add-step change_pass -r users -r appuser  
Adding sql/deploy/change_pass.sql  
-- requires: users, appuser  
Adding sql/revert/change_pass.sql  
Adding sql/test/change_pass.sql  
  
> emacs sql/deploy/insert_user.sql  
>
```

A Twofer

```
Terminal  
> sqitch add-step insert_user -r users -r appuser  
Adding sql/deploy/insert_user.sql  
-- requires: users, appuser  
Adding sql/revert/insert_user.sql  
Adding sql/test/insert_user.sql  
  
> sqitch add-step change_pass -r users -r appuser  
Adding sql/deploy/change_pass.sql  
-- requires: users, appuser  
Adding sql/revert/change_pass.sql  
Adding sql/test/change_pass.sql  
  
> emacs sql/deploy/insert_user.sql  
> emacs sql/deploy/change_pass.sql
```

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 code inside the window is as follows:

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

-- XXX Add DDLs here.

COMMIT;
```

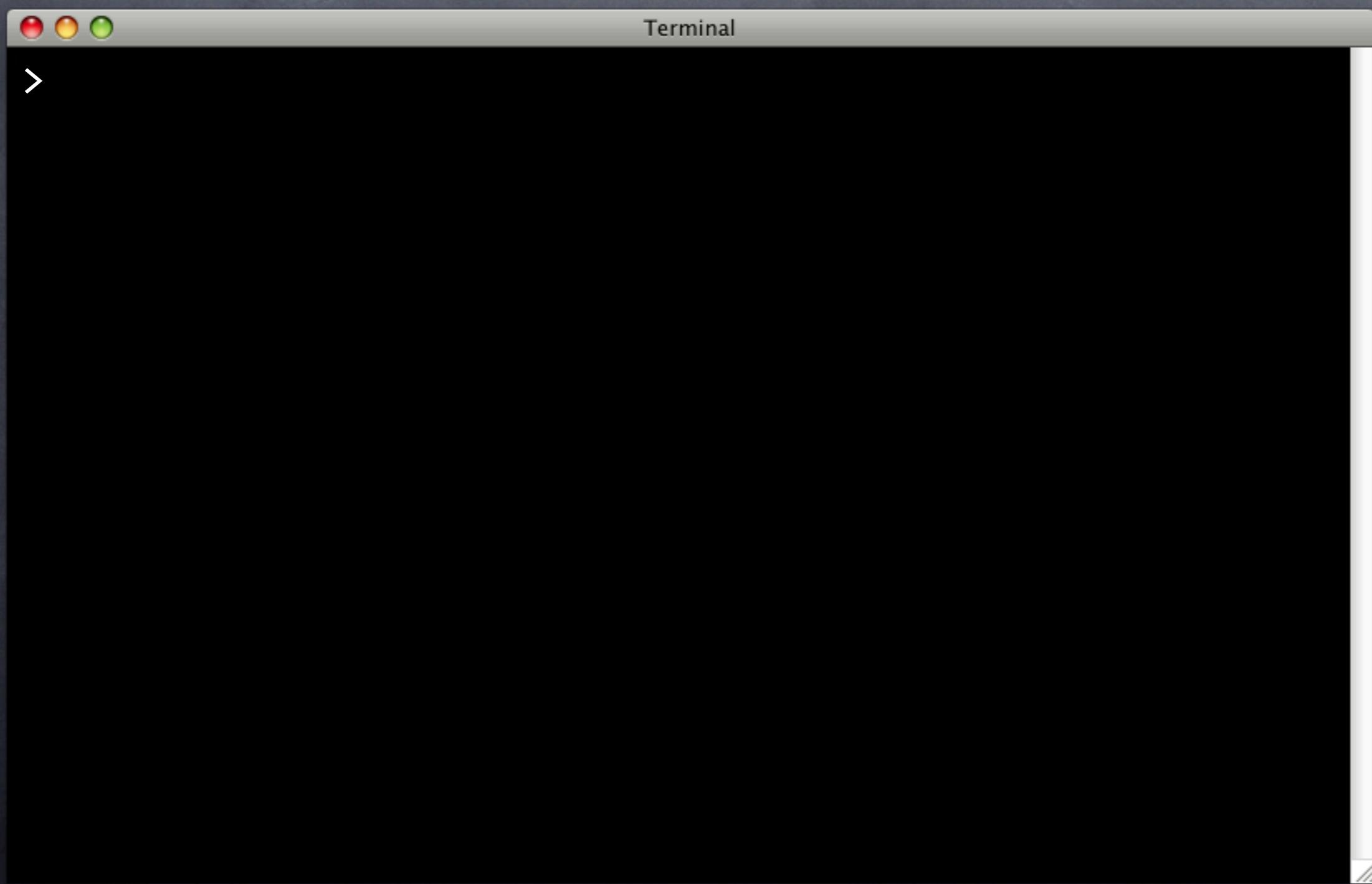
At the bottom of the window, there is a status bar with the text "---- sql/deploy/chan All (SQL[ansi])-----".

sql/deploy/ change_pass.sql

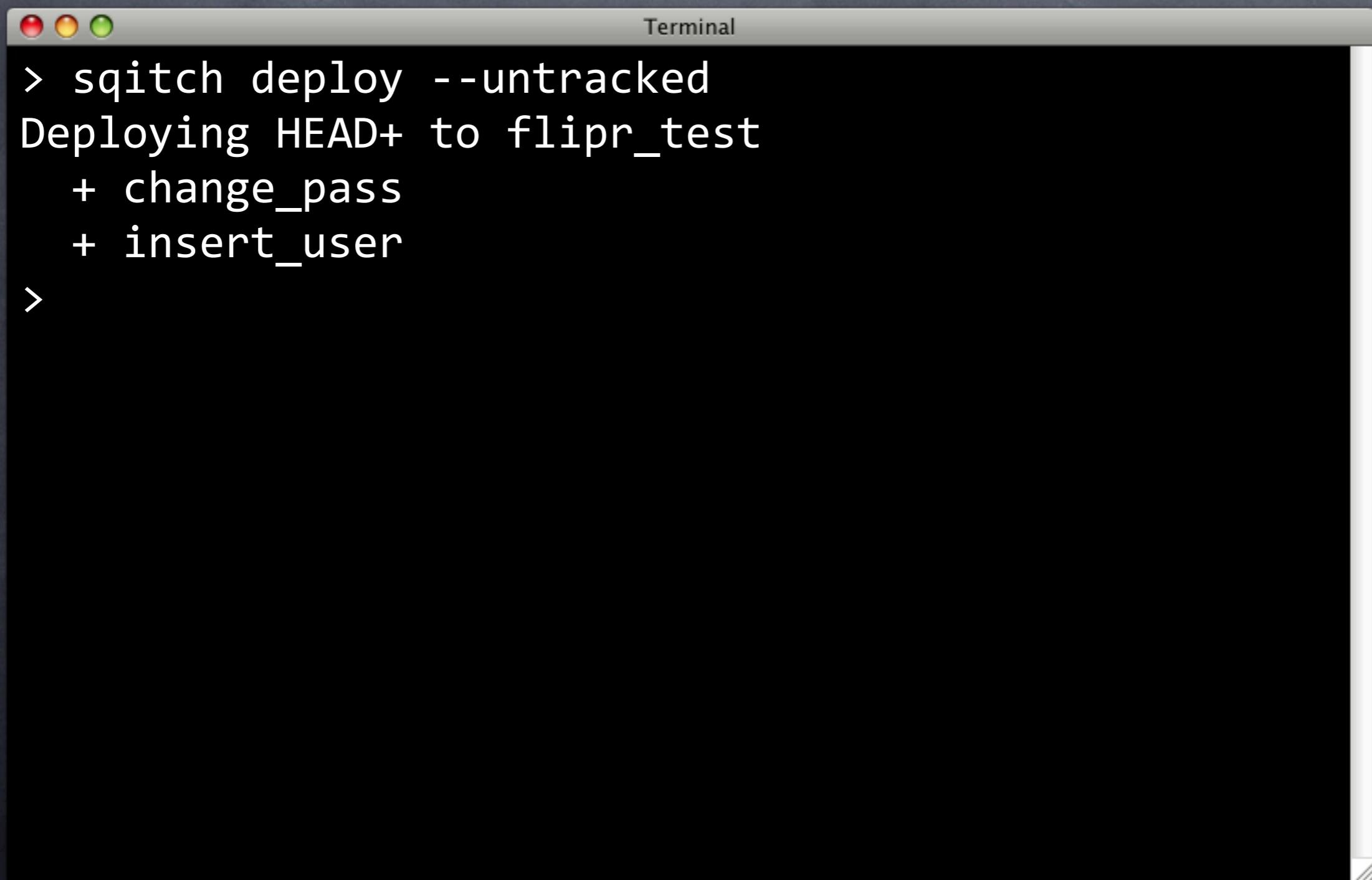
```
-- Deploy change_pass
-- :requires: appuser
-- :requires: users
BEGIN;
CREATE OR REPLACE FUNCTION change_pass(
    nick TEXT, oldpass TEXT, newpass TEXT
) RETURNS BOOLEAN LANGUAGE plpgsql SECURITY DEFINER AS $$%
BEGIN
    UPDATE users SET password = md5($3)
        WHERE nickname = $1 AND password = md5($2);
    RETURN FOUND;
END;
$$%
GRANT EXECUTE ON FUNCTION change_pass(TEXT, TEXT, TEXT) TO
flipr;
COMMIT;

--- sql/deploy/chan All (SQL[ansi])---
```

Deploy Functions



Deploy Functions



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 --untracked
Deploying HEAD+ to flipr_test
  + change_pass
  + insert_user
>
```

Deploy Functions

```
Terminal
> sqitch deploy --untracked
Deploying HEAD+ to flipr_test
  + change_pass
  + insert_user
> psql -d flipr_test -c '\df'                                List of function
Schema |      Name       | Result data type |
-----+-----+-----+-----+
public | change_pass | boolean          | nick text
public | insert_user | void            | nickname
>
```

Commit It

Commit It

- `sqitch revert --to HEAD`

Commit It

- `sqitch revert --to HEAD`
- `git add && git commit`

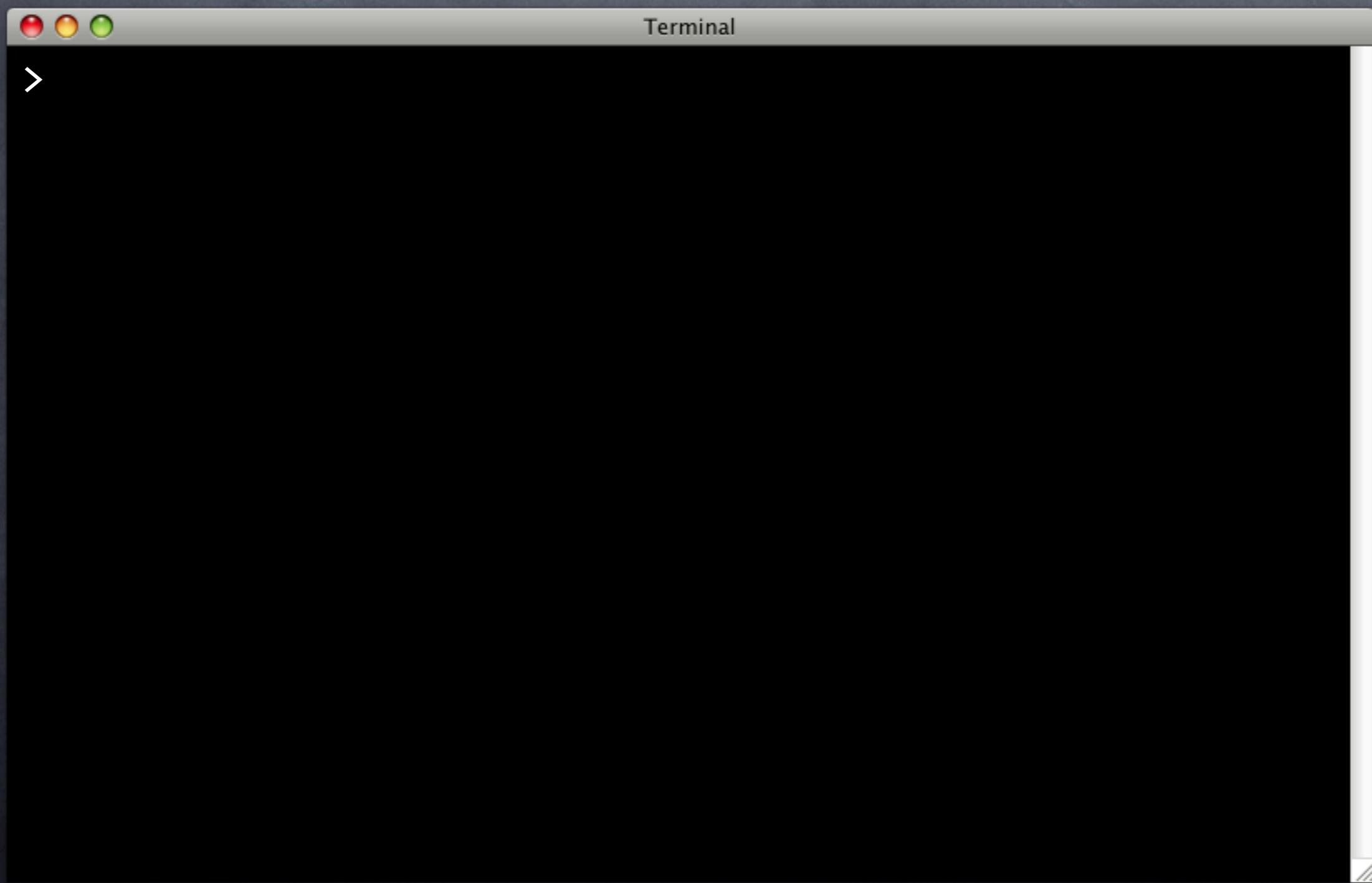
Commit It

- `sqitch revert --to HEAD`
- `git add && git commit`
- `sqitch deploy`

Commit It

- `sqitch revert --to HEAD`
- `git add && git commit`
- `sqitch deploy`
- `And now...`

Stage It



Stage It

```
Terminal
> git tag v1.0.0-dev1 -am 'Tag v1.0.0-dev1'
```

Stage It

```
Terminal  
> git tag v1.0.0-dev1 -am 'Tag v1.0.0-dev1'  
> createdb flipr_dev  
> sqitch --db-name flipr_dev deploy  
Deploying 36acafcd to flipr_dev  
  + appuser  
Deploying fa650af to flipr_dev  
  + users  
Deploying 803e6b8, v1.0.0-dev1 to flipr_dev  
  + insert_user  
  + change_pass  
>
```

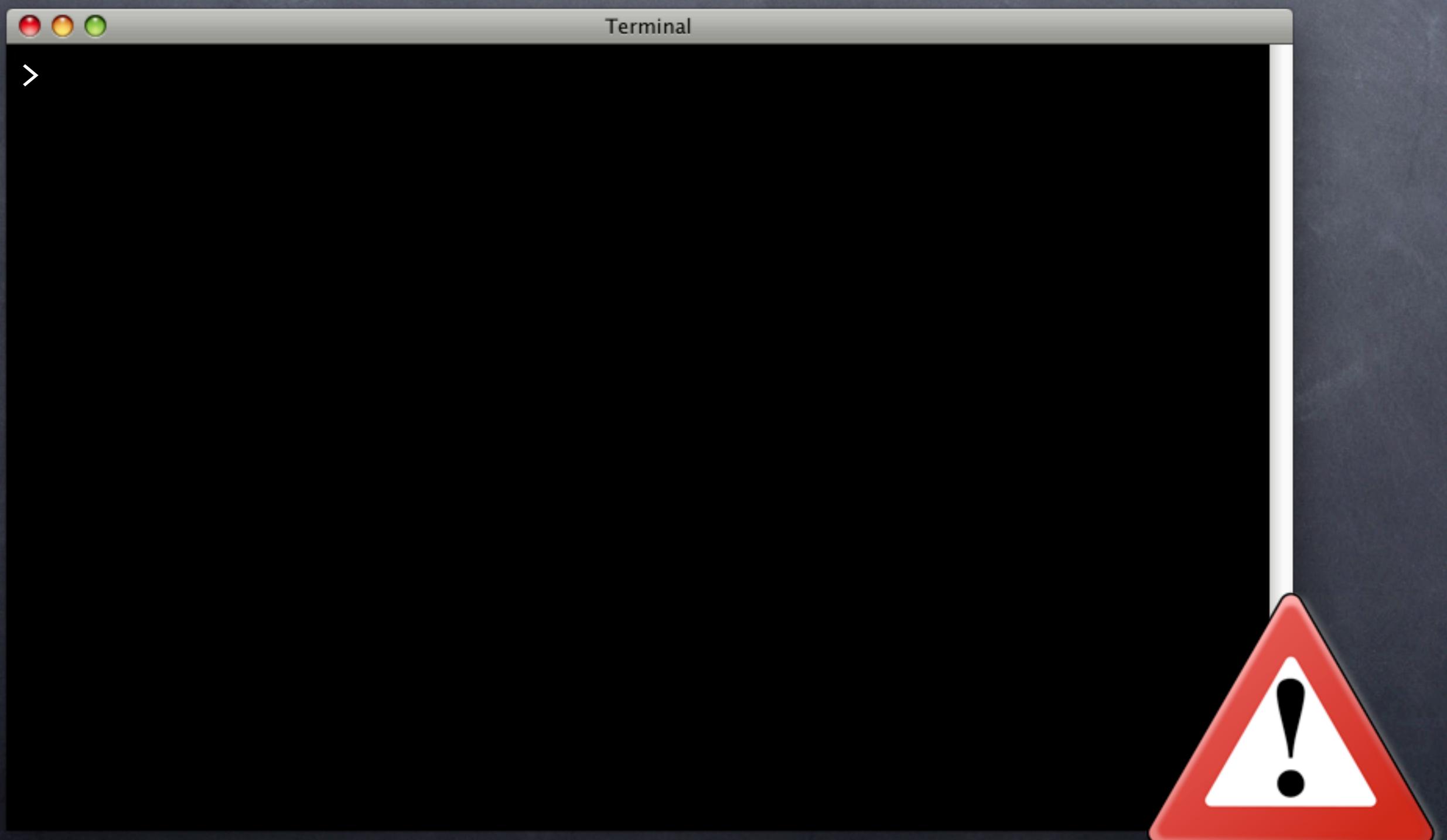
Stage It

```
Terminal  
> git tag v1.0.0-dev1 -am 'Tag v1.0.0-dev1'  
> createdb flipr_dev  
> sqitch --db-name flipr_dev deploy  
Deploying 36acafcd to flipr_dev  
+ appuser  
Deploying fa650af to flipr_dev  
+ users  
Deploying 803e6b8, v1.0.0-dev1 to flipr_dev  
+ insert_user  
+ change_pass  
>
```

Stage It

```
Terminal  
> git tag v1.0.0-dev1 -am 'Tag v1.0.0-dev1'  
> createdb flipr_dev  
> sqitch --db-name flipr_dev deploy  
Deploying 36acafcd to flipr_dev  
  + appuser  
Deploying fa650af to flipr_dev  
  + users  
Deploying 803e6b8, v1.0.0-dev1 to flipr_dev  
  + insert_user  
  + change_pass  
>
```

Stage Status?



Stage Status?

```
Terminal  
> sqitch --db-name flipr_dev status --show tags  
# On database flipr_dev  
# Tags: 803e6b8, v1.0.0-dev1  
# Step: change_pass  
# Date: 2012-05-18 17:38:28  
#  
# Tags:  
#     803e6b8, v1.0.0-dev1 - 2012-05-18 17:38:28 - david  
#     fa650af                 - 2012-05-18 17:38:21 - david  
#     36acafdf                - 2012-05-18 17:38:15 - david  
#  
Nothing to deploy (up-to-date)  
>
```



Distribution Bundling

Distribution Bundling

- Bundle plan + scripts for deployment

Distribution Bundling

- Bundle plan + scripts for deployment
- Plan generated from VCS history

Distribution Bundling

- Bundle plan + scripts for deployment
- Plan generated from VCS history
- Package up as

Distribution Bundling

- Bundle plan + scripts for deployment
- Plan generated from VCS history
- Package up as
 - RPM

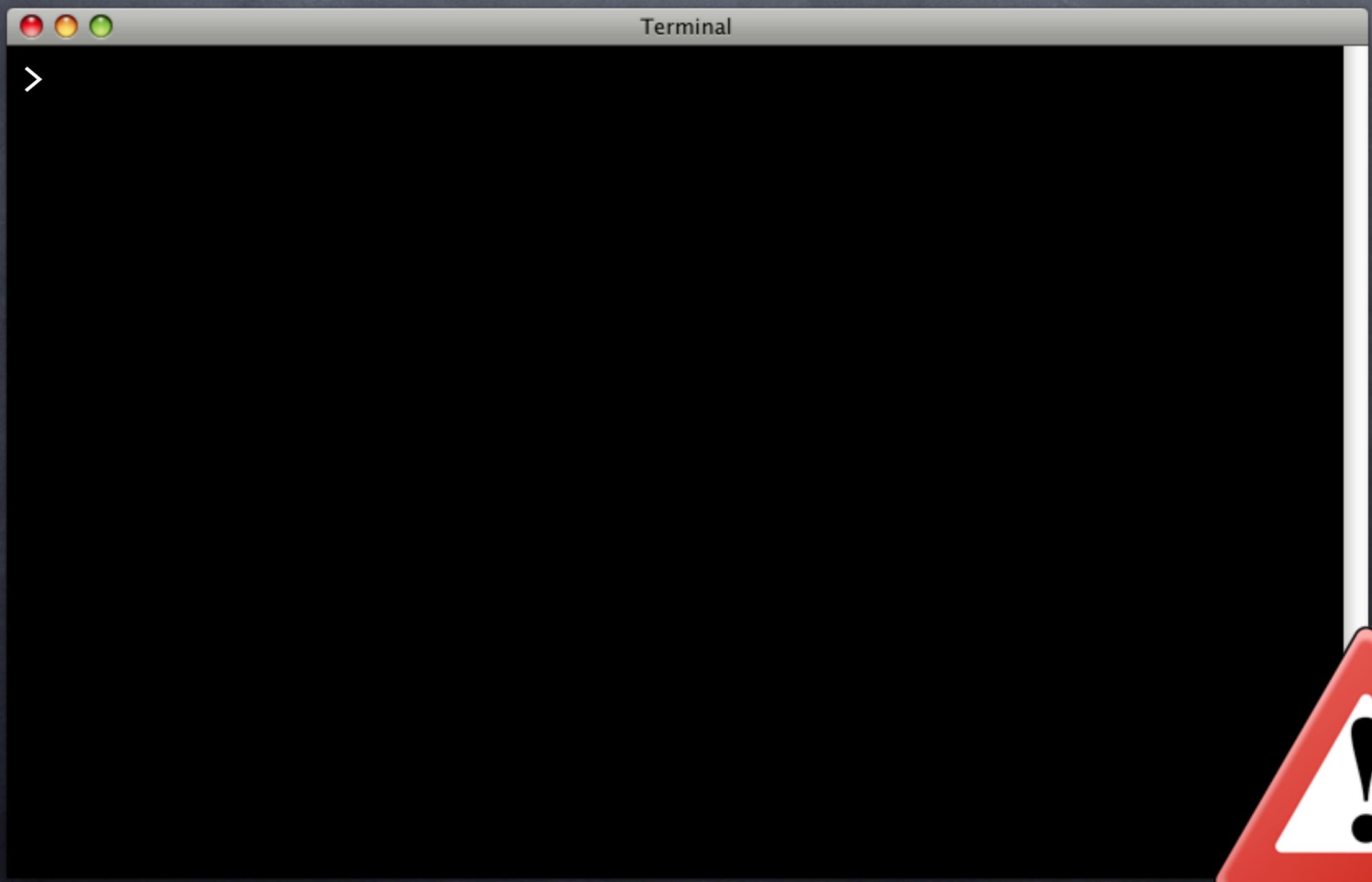
Distribution Bundling

- ⦿ Bundle plan + scripts for deployment
- ⦿ Plan generated from VCS history
- ⦿ Package up as
 - ⦿ RPM
 - ⦿ Debian Package

Distribution Bundling

- Bundle plan + scripts for deployment
- Plan generated from VCS history
- Package up as
 - RPM
 - Debian Package
 - Gem

Ship It!



Ship It!

```
Terminal  
> sqitch bundle --tags-only  
Bundling in bundle/  
Config written to bundle/sqitch.conf  
Bundling v1.0.0-dev1  
- appuser  
- users  
- change_pass  
- insert_user  
Plan written to bundle/sqitch.plan  
>
```



Ship It!

```
Terminal  
> sqitch bundle --tags-only  
Bundling in bundle/  
Config written to bundle/sqitch.conf  
Bundling v1.0.0-dev1  
- appuser  
- users  
- change_pass  
- insert_user  
Plan written to bundle/sqitch.plan  
>
```



Ship It!

```
Terminal  
> sqitch bundle --tags-only  
Bundling in bundle/  
Config written to bundle/sqitch.conf  
Bundling v1.0.0-dev1  
- appuser  
- users  
- change_pass  
- insert_user  
Plan written to bundle/sqitch.plan  
>
```



Ship It!

```
Terminal  
> sqitch bundle --tags-only  
Bundling in bundle/  
Config written to bundle/sqitch.conf  
Bundling v1.0.0-dev1  
- appuser  
- users  
- change_pass  
- insert_user  
Plan written to bundle/sqitch.plan  
>
```

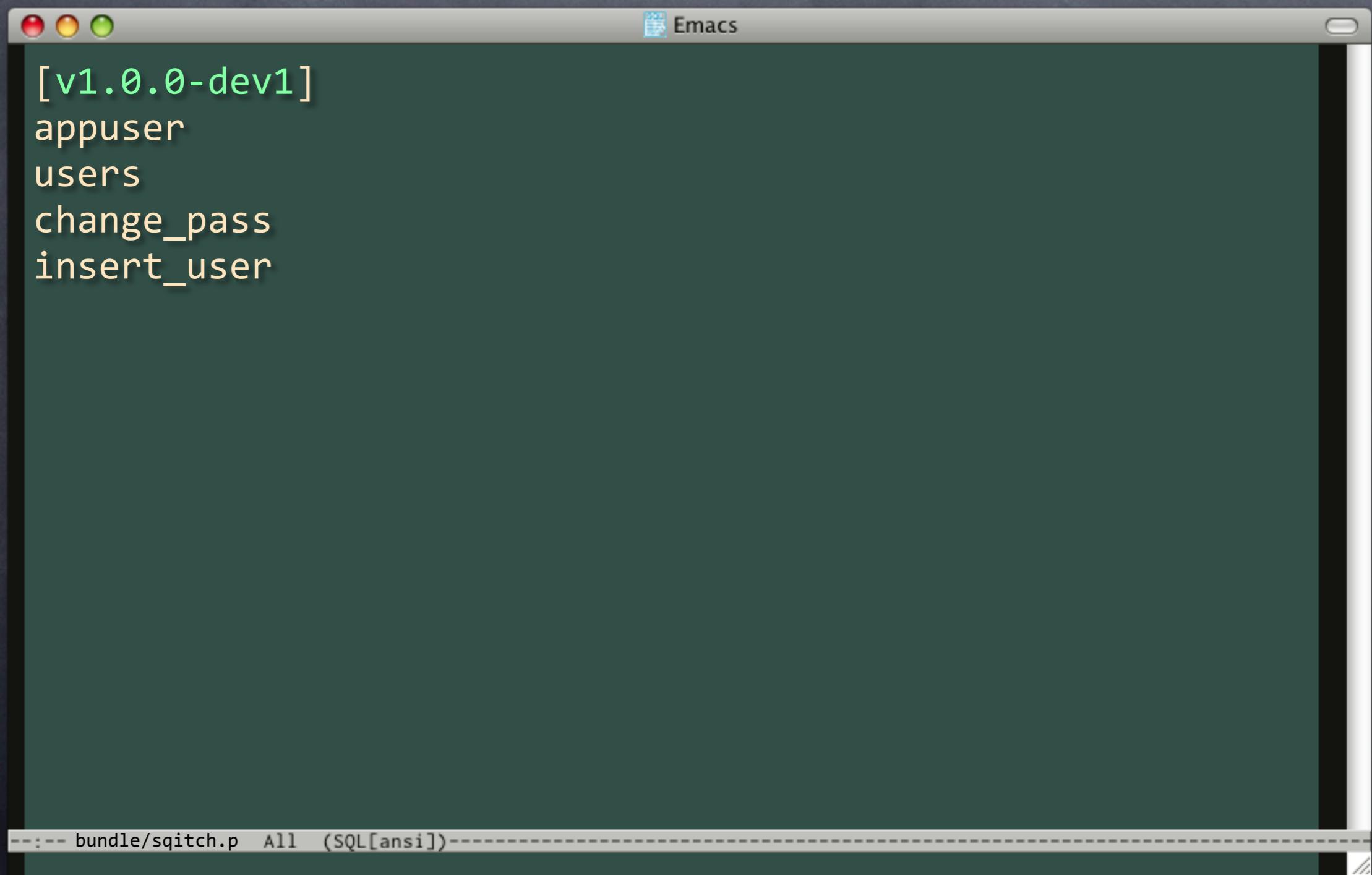


Ship It!

```
Terminal  
> sqitch bundle --tags-only  
Bundling in bundle/  
Config written to bundle/sqitch.conf  
Bundling v1.0.0-dev1  
- appuser  
- users  
- change_pass  
- insert_user  
Plan written to bundle/sqitch.plan  
> emacs bundle/sqitch.plan
```



The Plan, Man



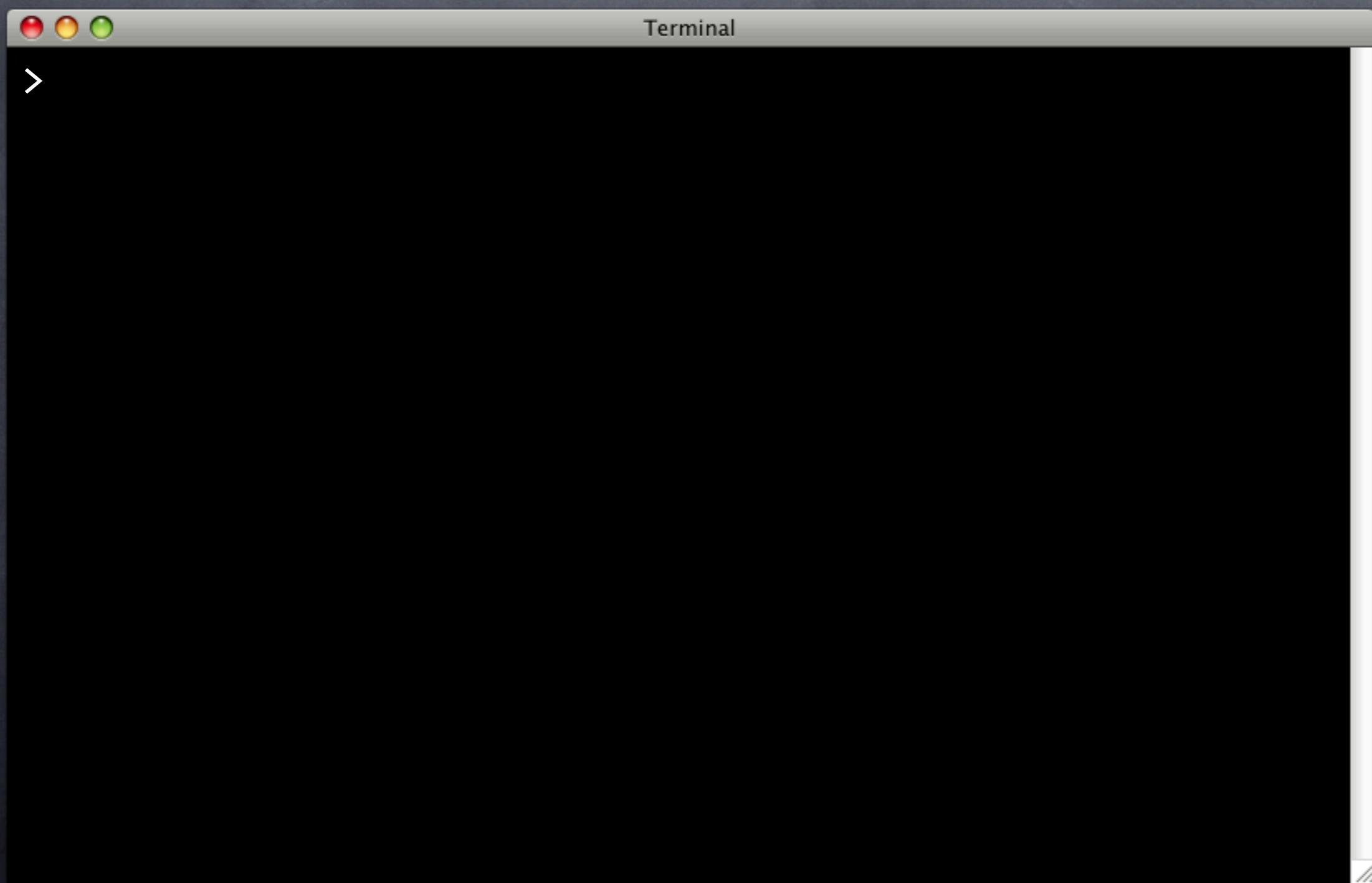
A screenshot of an Emacs window titled "Emacs". The window has a dark green background and contains a list of database schema changes. The text is white and includes:

```
[v1.0.0-dev1]
appuser
users
change_pass
insert_user
```

At the bottom of the window, there is a status bar with the text:

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

Kick the Tires



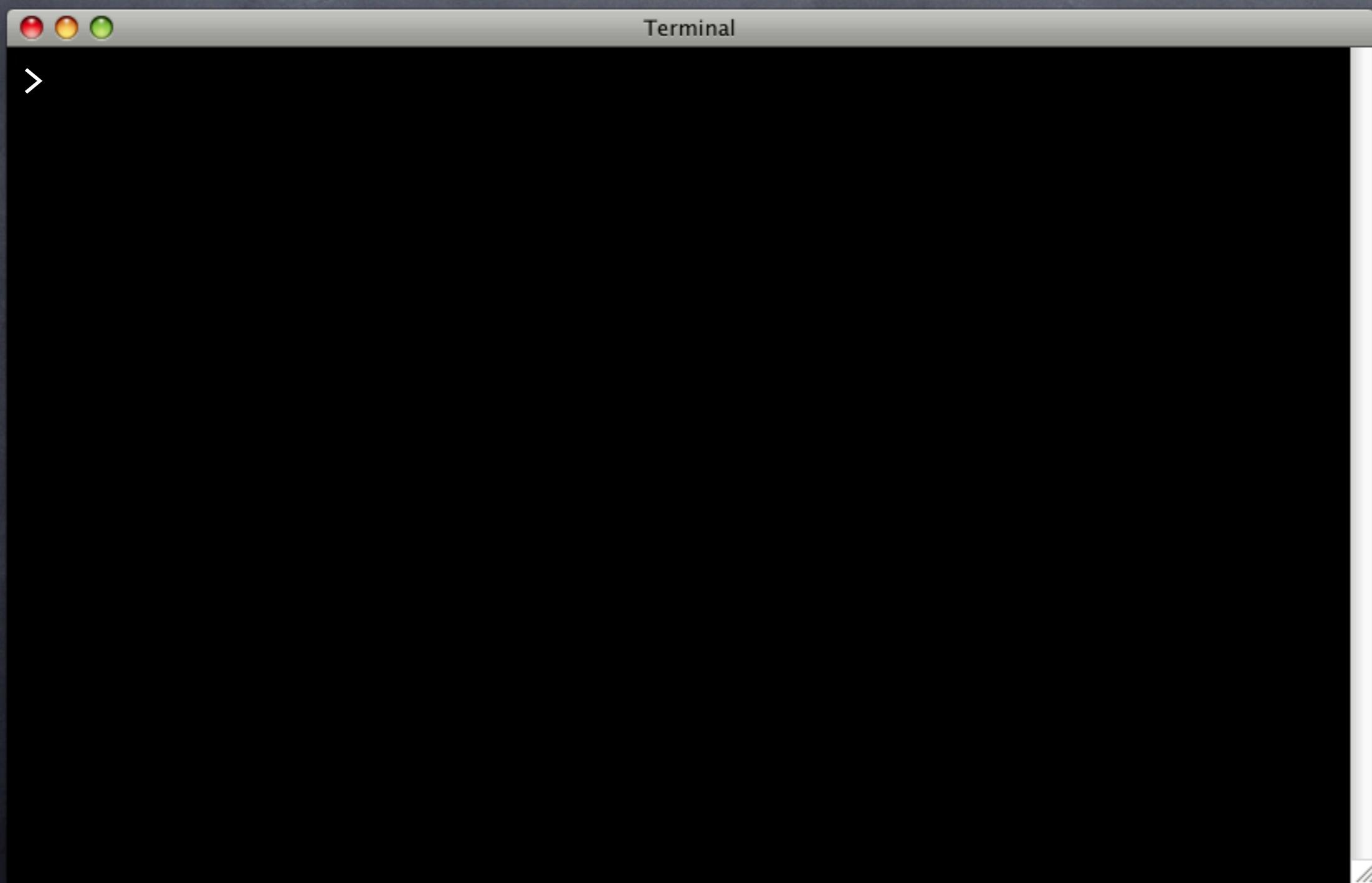
Kick the Tires

```
Terminal
> cd bundle
> createdb flipr_staging
> sqitch --db-name flipr_staging deploy
Deploying v1.0.0-dev1 to flipr_prod
+ appuser
+ users
+ insert_user
+ change_pass
>
```

Kick the Tires

```
Terminal
> cd bundle
> createdb flipr_staging
> sqitch --db-name flipr_staging deploy
Deploying v1.0.0-dev1 to flipr_prod
+ appuser
+ users
+ insert_user
+ change_pass
>
```

Ch-Check it Out



Ch-Check it Out



Terminal

```
> sqitch --db-name flipr_prod status --show tags
# On database flipr_prod
# Tag: v1.0.0-dev1
# Step: change_pass
# Date: 2012-04-09 22:17:38
#
# Tags:
#     v1.0.0-dev1 - 2012-05-18 17:40:00 - david
#
Nothing to deploy (up-to-date)
>
```

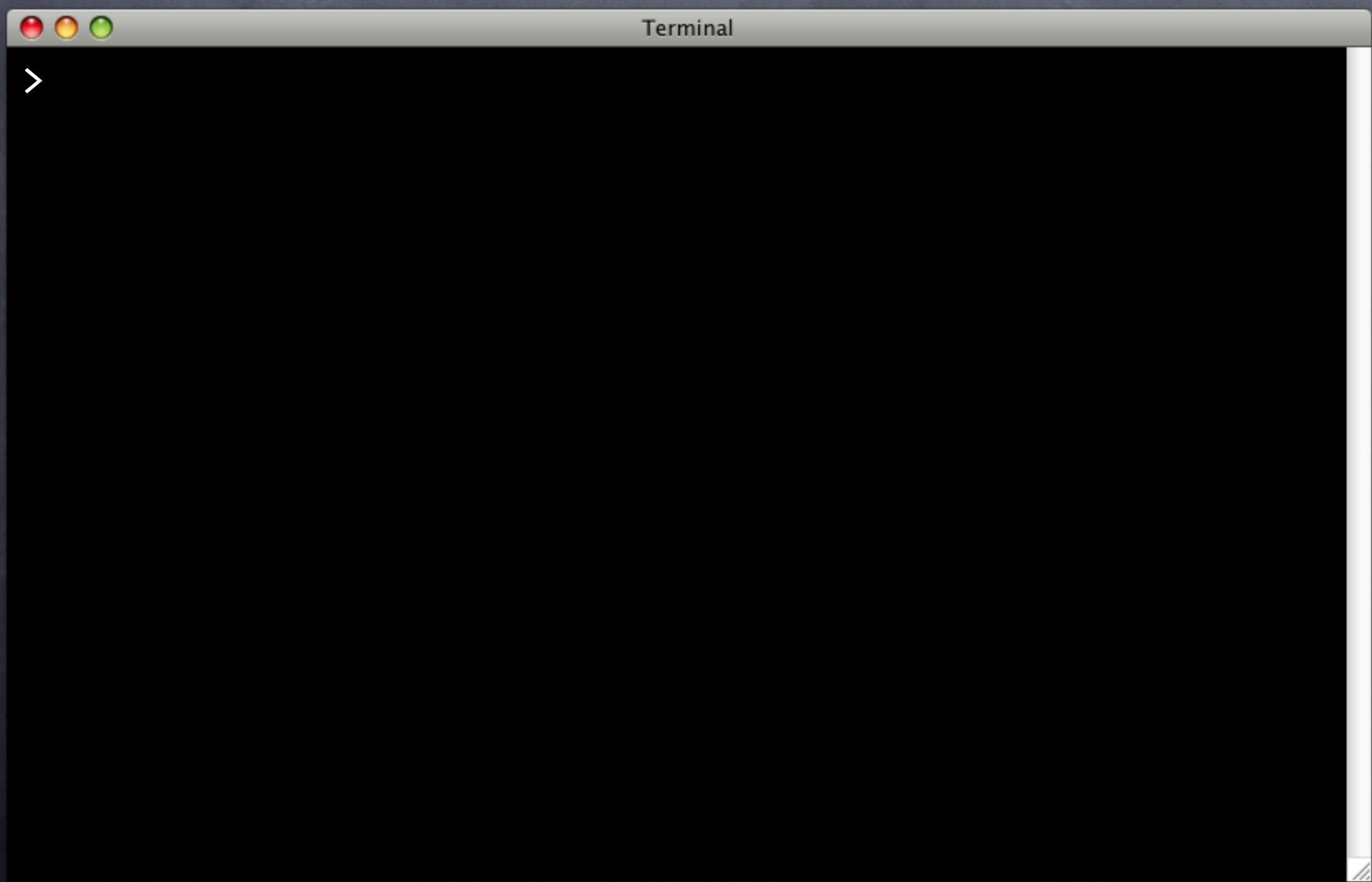
Ch-Check it Out

```
Terminal  
> sqitch --db-name flipr_prod status --show tags  
# On database flipr_prod  
# Tag: v1.0.0-dev1  
# Step: change_pass  
# Date: 2012-04-09 22:17:38  
#  
# Tags:  
#      v1.0.0-dev1 - 2012-05-18 17:40:00 - david  
#  
Nothing to deploy (up-to-date)  
>
```



No SHA1s

Ruh-Roh



Ruh-Roh

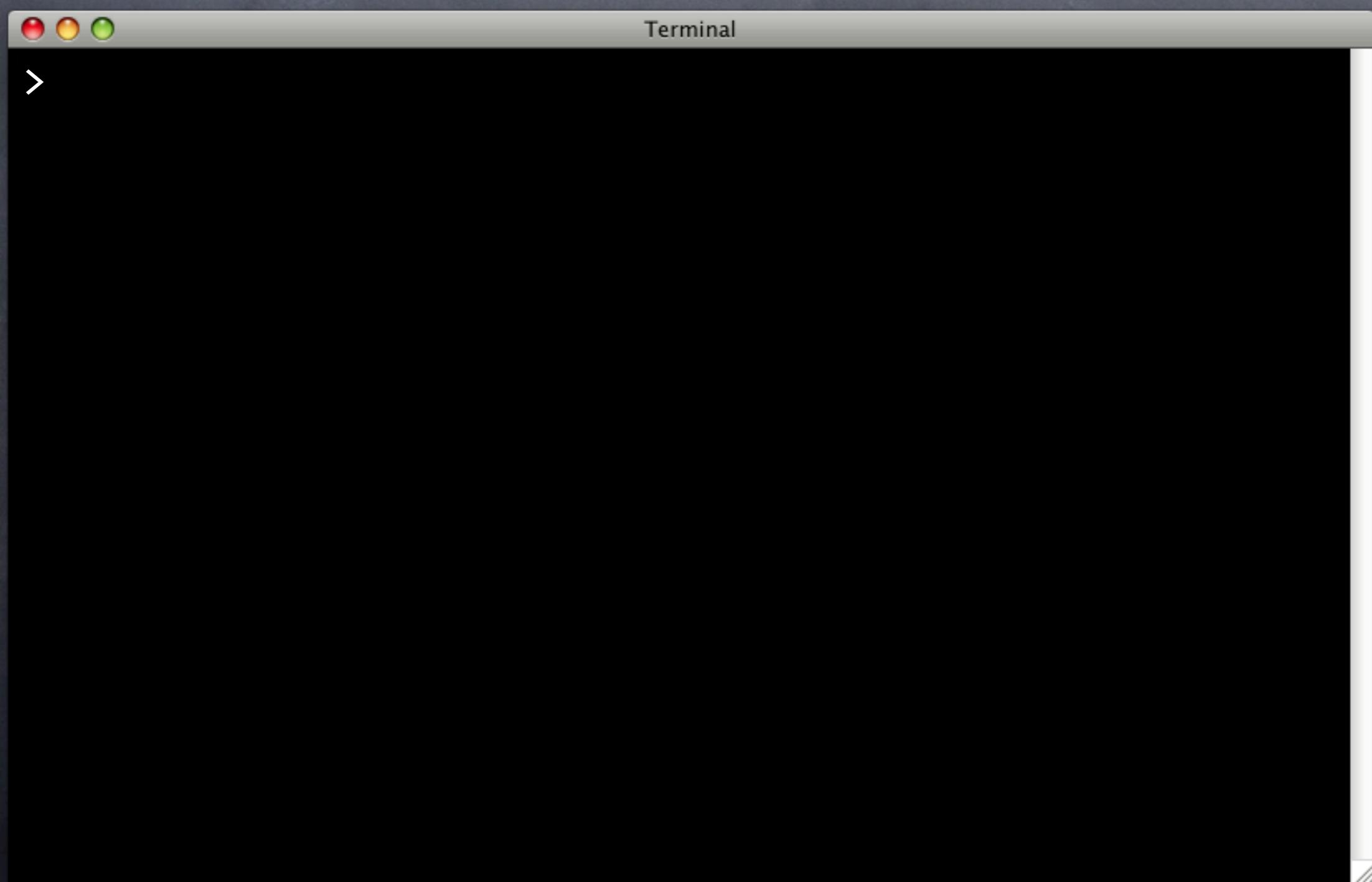
```
Terminal
> psql -d flipr_test -c "
    SELECT insert_user('foo', 'secr3t'),
           insert_user('bar', 'secr3t');
    SELECT nickname, password FROM users;
"
nickname | password
-----+-----
foo      | 9695da4dd567a19f9b92065f240c6725
bar      | 9695da4dd567a19f9b92065f240c6725
```

Ruh-Roh

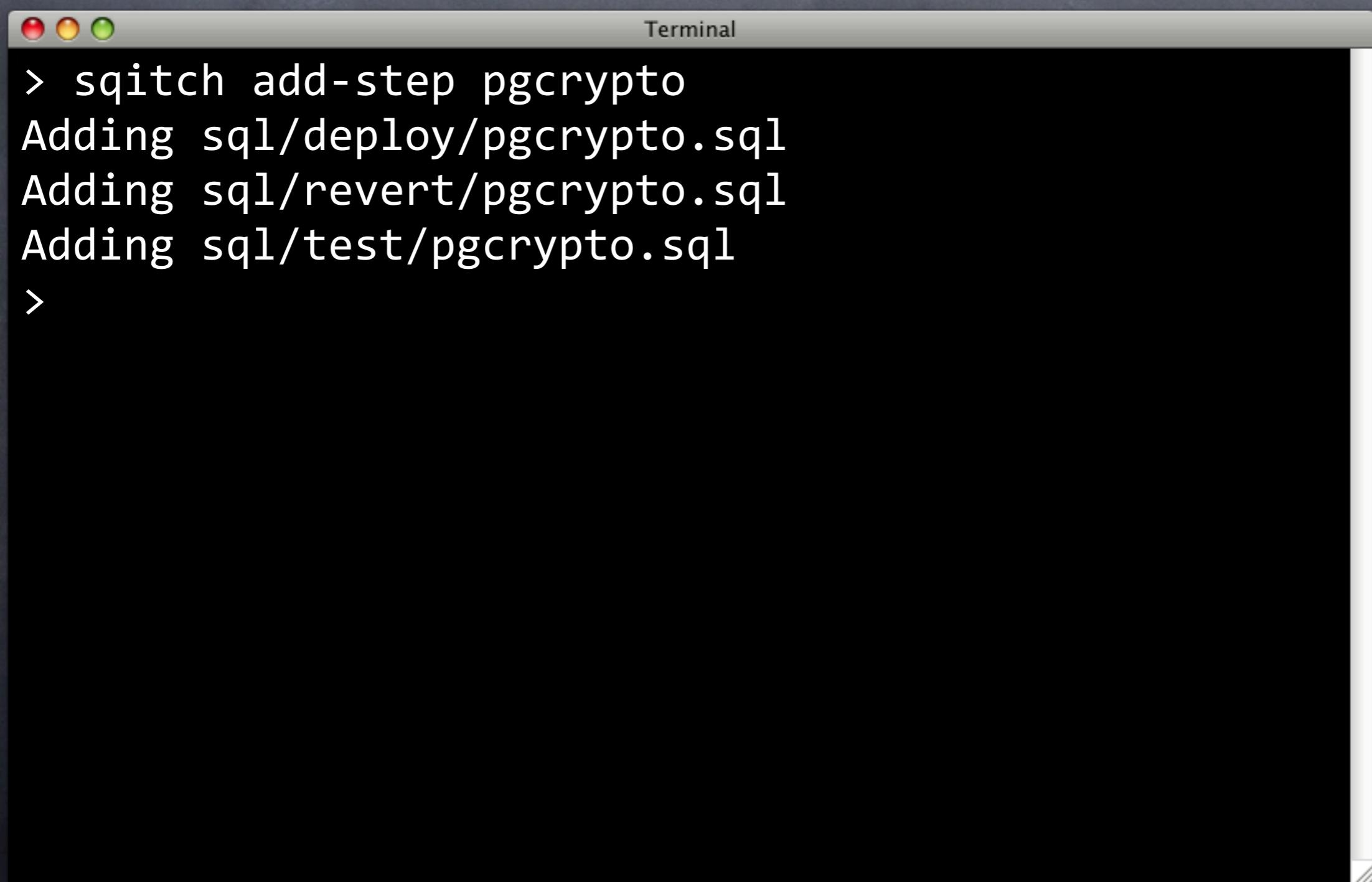
```
Terminal
> psql -d flipr_test -c "
    SELECT insert_user('foo', 'secr3t'),
           insert_user('bar', 'secr3t');
    SELECT nickname, password FROM users;
"
nickname | password
-----
foo      | 9695da4dd567a19f9b92065f240c6725
bar      | 9695da4dd567a19f9b92065f240c6725
```

Not good.

Add pgcrypto



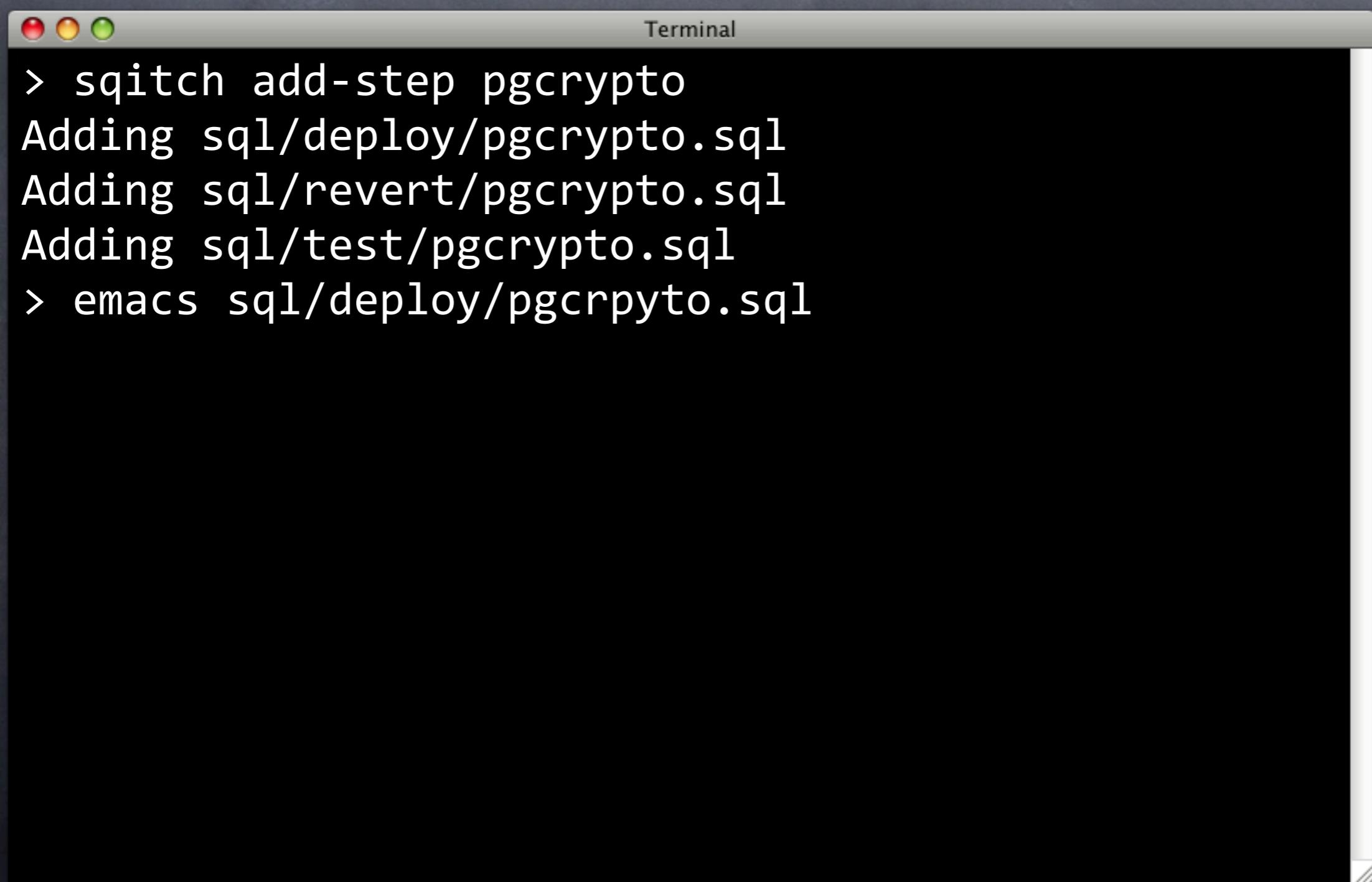
Add pgcrypto

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-step pgcrypto
Adding sql/deploy/pgcrypto.sql
Adding sql/revert/pgcrypto.sql
Adding sql/test/pgcrypto.sql
>
```

The terminal window is set against a dark background.

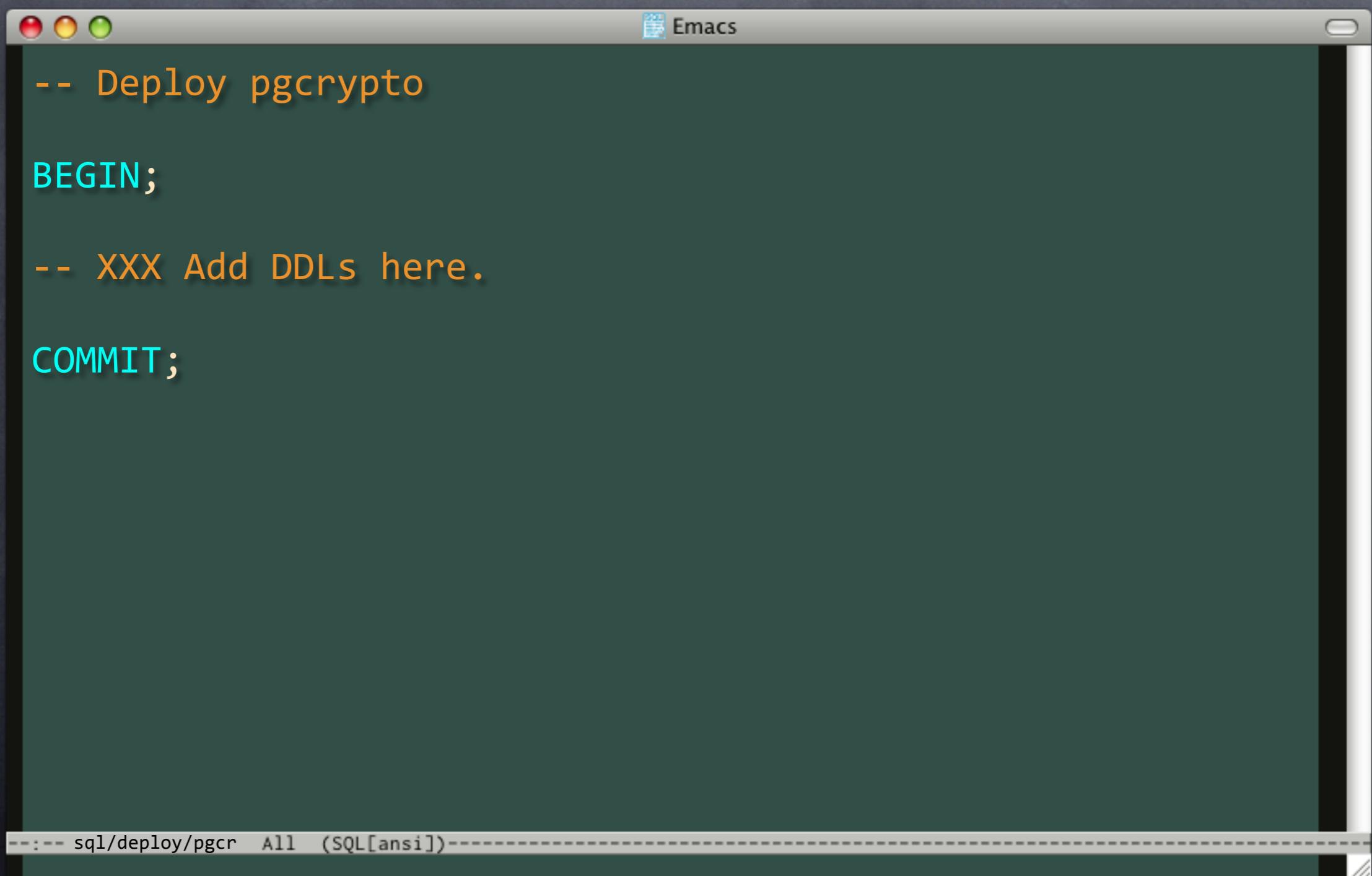
Add pgcrypto



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-step pgcrypto
Adding sql/deploy/pgcrypto.sql
Adding sql/revert/pgcrypto.sql
Adding sql/test/pgcrypto.sql
> emacs sql/deploy/pgcrypto.sql
```

ssql/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

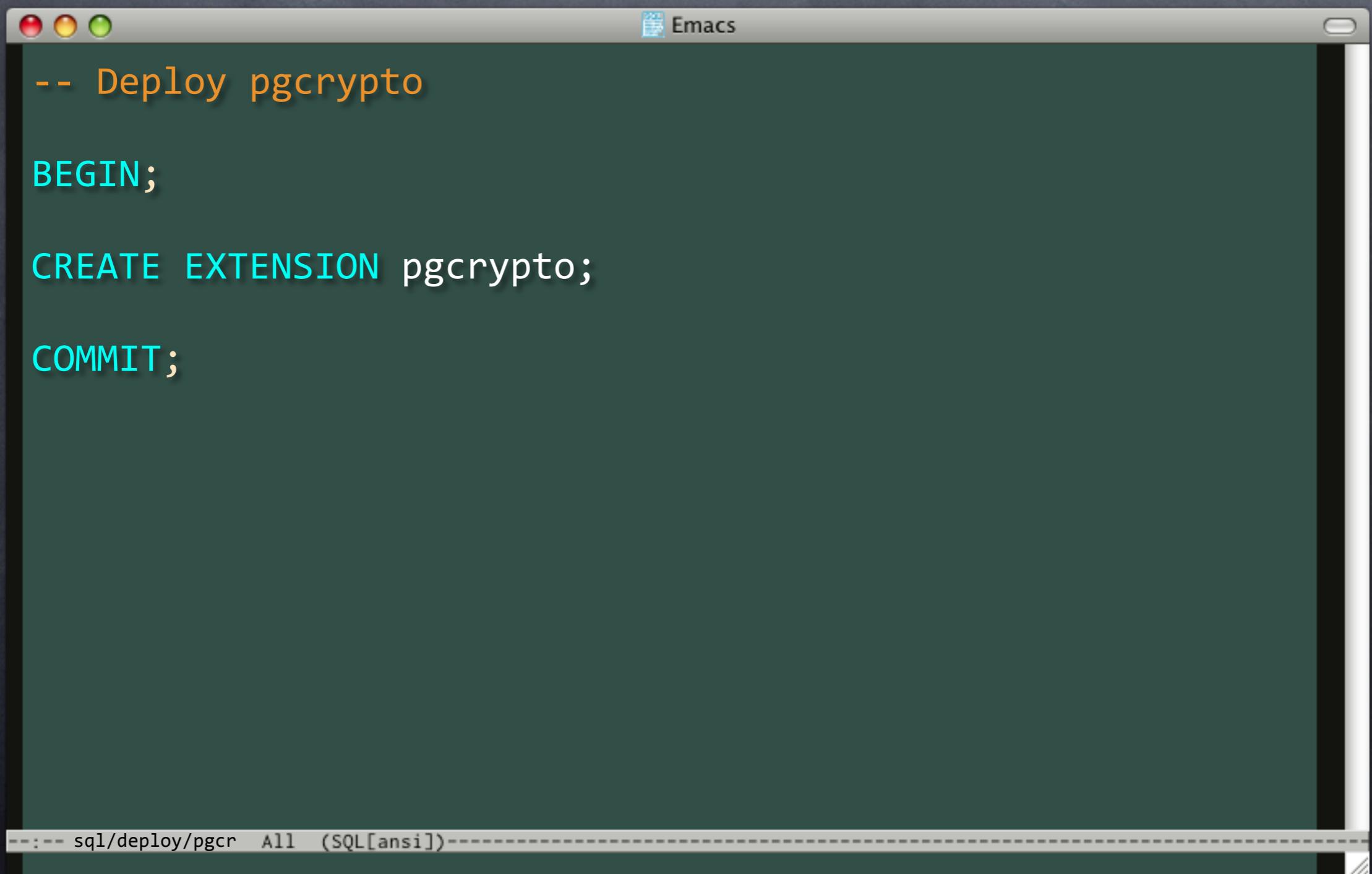
BEGIN;

-- XXX Add DDLs here.

COMMIT;
```

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

ssql/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 "----:--- sql/deploy/pgcr 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/sql/deploy/insert_user_crypt.sql b/sql/deploy/insert_user_cr
new file mode 100644
index 0000000..fa8d0c6
--- /dev/null
+++ b/sql/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/sql/revert/insert_user_crypt.sql b/sql/revert/insert_user_cr
new file mode 100644
index 0000000..a7f4e31
--- /dev/null
+++ b/sql/revert/insert_user_crypt.sql
@@ -0,0 +1,8 @@
+-- requires: users, appuser
+
+CREATE OR REPLACE FUNCTION insert_user(
+    nickname TEXT,
+    password TEXT
+) RETURNS VOID LANGUAGE SQL AS $$
+    INSERT INTO users values($1, md5($2));
+$$;
```



Terminal

```
> git diff HEAD^
diff --git a/sql/deploy/insert_user_crypt.sql b/sql/deploy/insert_user_cr
new file mode 100644
index 0000000..fa8d0c6
--- /dev/null
+++ b/sql/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/sql/revert/insert_user_crypt.sql b/sql/revert/insert_user_cr
new file mode 100644
index 0000000..a7f4e31
--- /dev/null
+++ b/sql/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.

Use the VCS, Luke

Use the VCS, Luke

- Just modify deploy scripts directly

Use the VCS, Luke

- ⦿ Just modify deploy scripts directly
- ⦿ Requirement:

Use the VCS, Luke

- Just modify deploy scripts directly
- Requirement:
 - Change must be idempotent

Use the VCS, Luke

- ⦿ Just modify deploy scripts directly
- ⦿ Requirement:
 - ⦿ Change must be idempotent
 - ⦿ Same result no matter how often applied

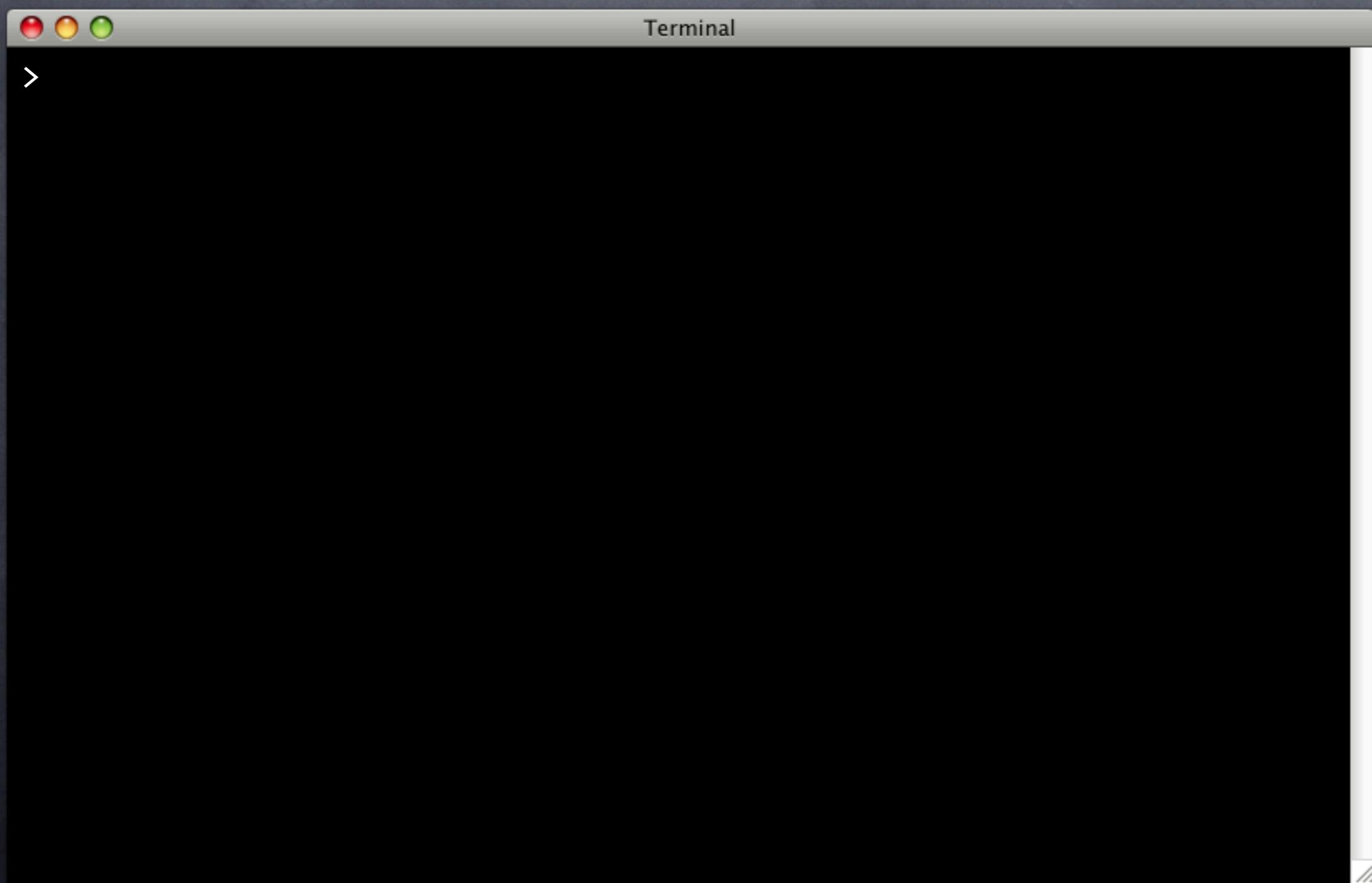
Use the VCS, Luke

- Just modify deploy scripts directly
- Requirement:
 - Change must be idempotent
 - Same result no matter how often applied
 - CREATE OR REPLACE FUNCTION is idempotent

Use the VCS, Luke

- Just modify deploy scripts directly
- Requirement:
 - Change must be idempotent
 - Same result no matter how often applied
- CREATE OR REPLACE FUNCTION is idempotent
- So change procedures in place!

What's the Diff?

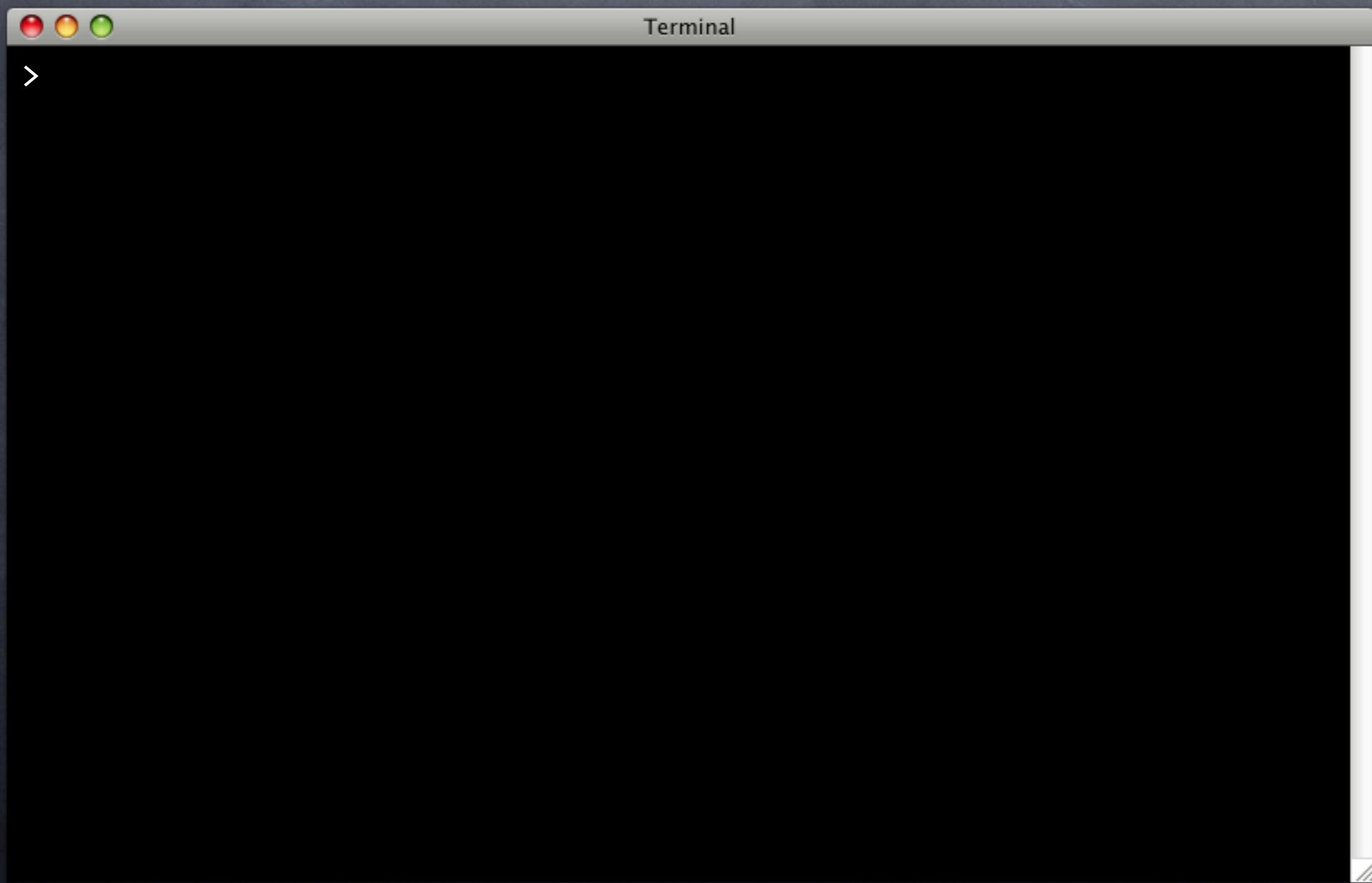


What's the Diff?

```
Terminal
> diff -u sql/deploy/insert_user.sql
@@ -1,8 +1,8 @@
--- requires: users, appuser
++ requires: users, appuser, pgcrypto

CREATE OR REPLACE FUNCTION insert_user(
    nickname TEXT,
    password TEXT
) RETURNS VOID LANGUAGE SQL AS $$
-    INSERT INTO users values($1, md5($2));
+    INSERT INTO users values($1, crypt($2, gen_salt('md5'))));
$$;
```

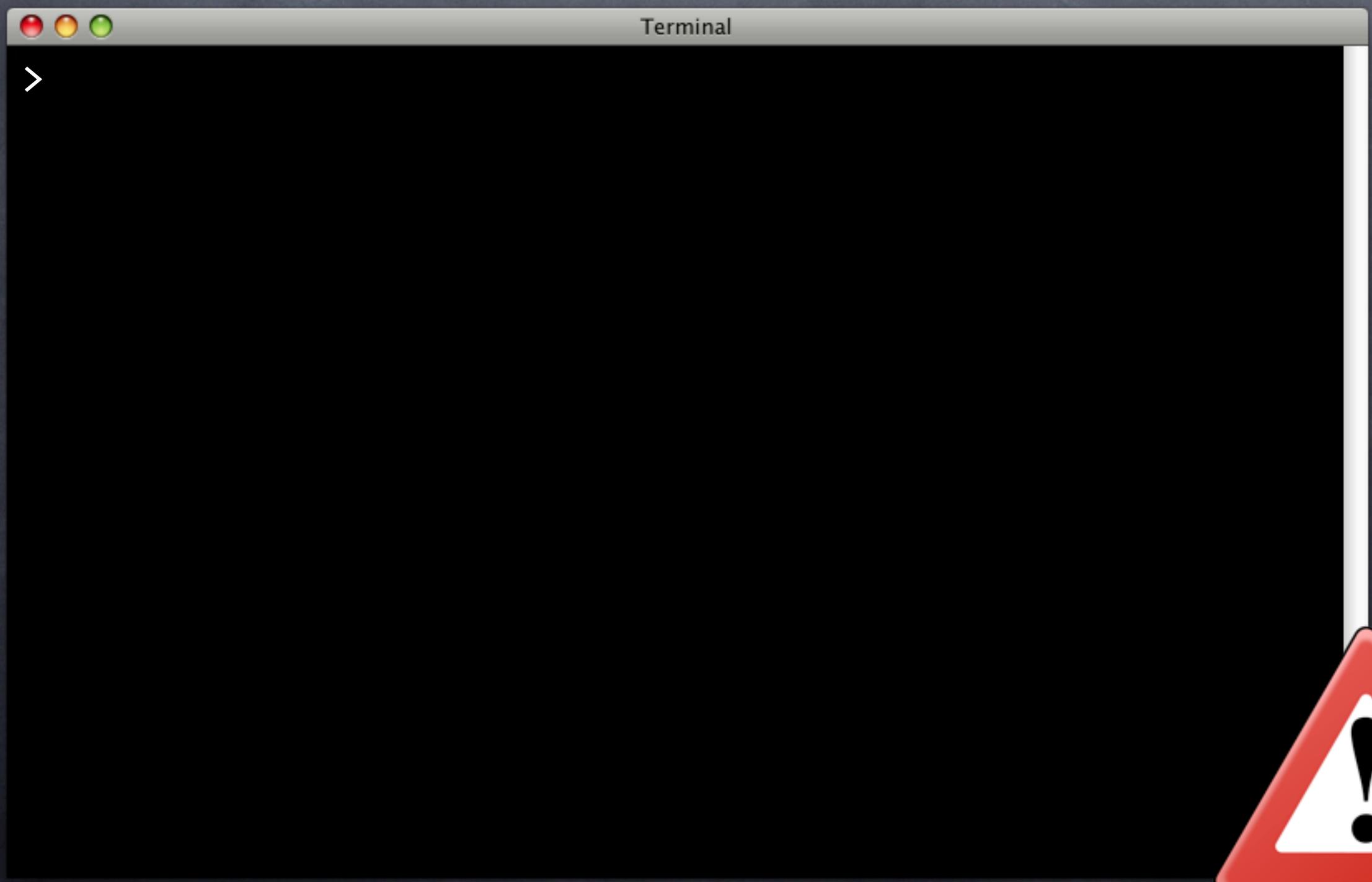
What's the Diff?



What's the Diff?

```
Terminal  
> diff -u sql/deploy/change_pass.sql  
@@ -1,4 +1,4 @@  
--- requires: users, appuser  
++ requires: users, appuser, pgcrypto  
  
CREATE OR REPLACE FUNCTION change_pass(  
    nick      TEXT,  
@@ -7,9 +7,9 @@ CREATE OR REPLACE FUNCTION change_pass(  
) RETURNS BOOLEAN LANGUAGE plpgsql AS $$  
BEGIN  
    UPDATE users  
    -        SET password = md5($3)  
    +        SET password = crypt($3, gen_salt('md5'))  
    WHERE nickname = $1  
    -        AND password = md5($2);  
    +        AND password = crypt($2, password);  
    RETURN FOUND;  
END;
```

Send it Up!



Send it Up!

```
Terminal  
> sqitch deploy --untracked  
Deploying HEAD+ to flipr_test  
  + insert_user  
  + change_pass  
>
```

A red triangular warning sign with a black exclamation mark in the center, positioned in the bottom right corner of the terminal window.

Send it Up!

```
Terminal  
> sqitch deploy --untracked  
Deploying HEAD+ 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  
>
```



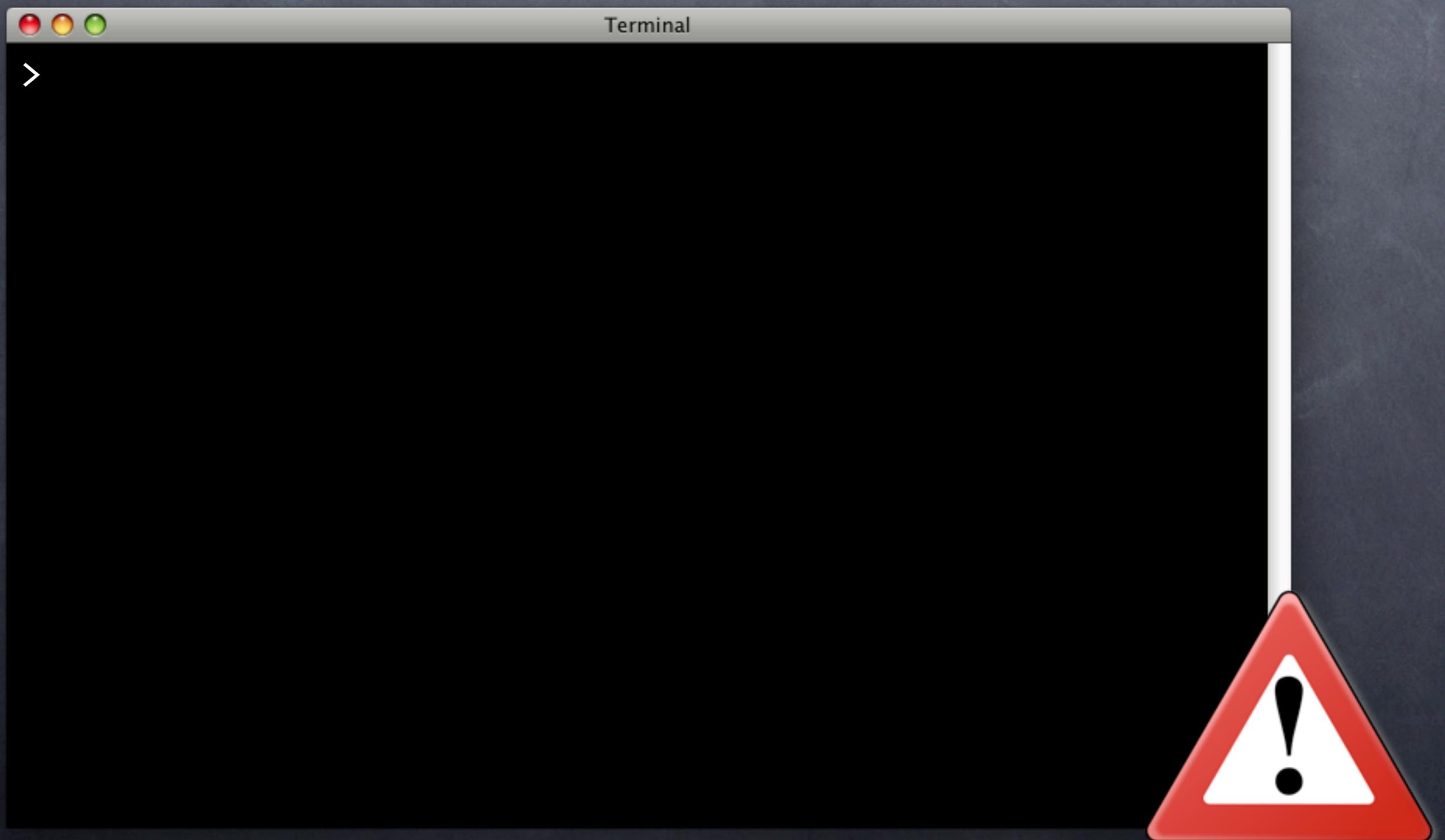
Send it Up!

```
Terminal  
> sqitch deploy --untracked  
Deploying HEAD+ 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  
>
```



A large white checkmark is positioned to the right of the terminal window.

Can We Go Back?



Can We Go Back?

```
Terminal  
> sqitch revert --to HEAD  
Reverting HEAD+ from flipr_test  
- change_pass  
- insert_user  
>
```

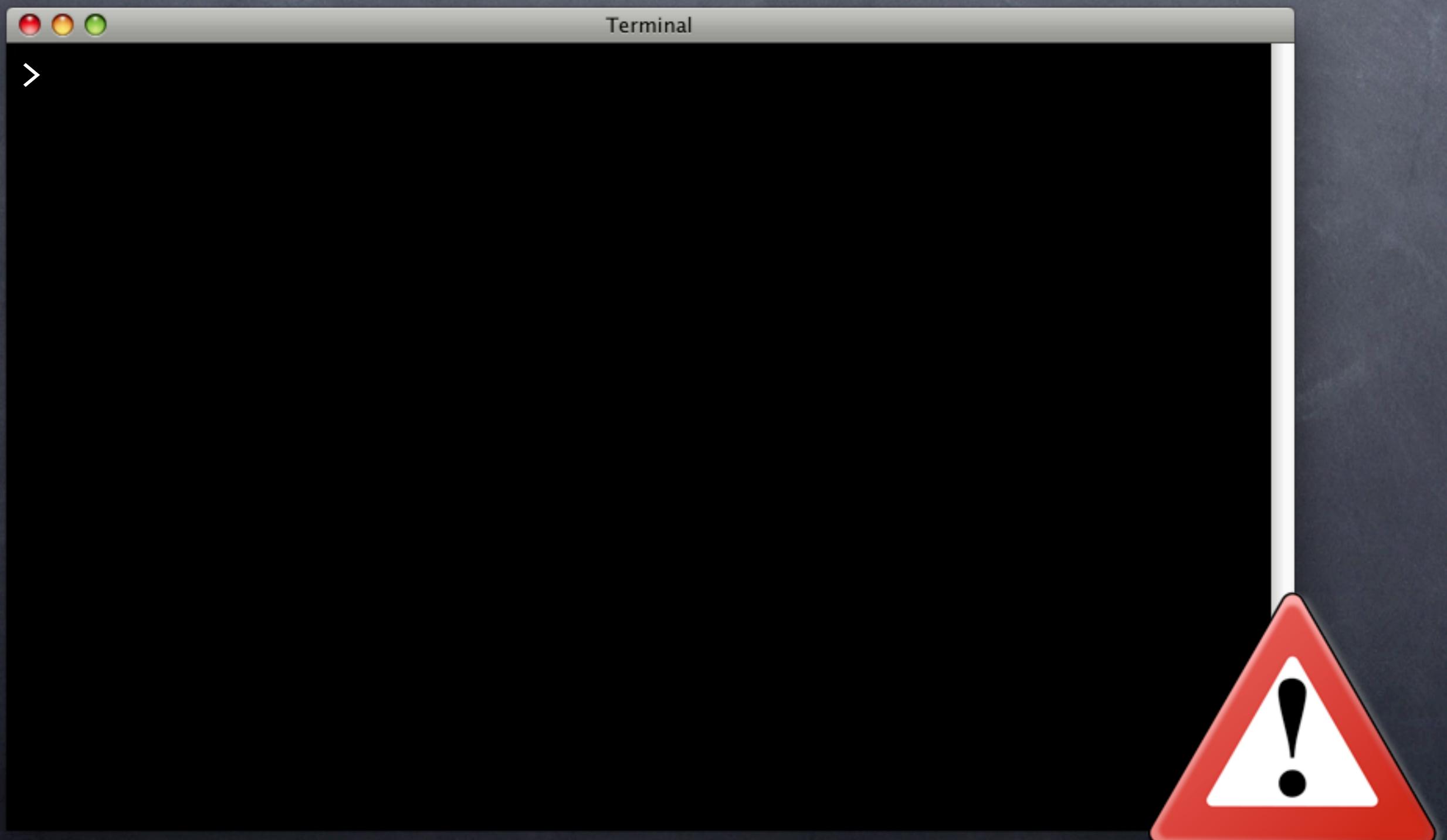
A red triangular warning sign with a black exclamation mark in the center, positioned in the bottom right corner of the terminal window.

Can We Go Back?

```
Terminal  
> sqitch revert --to HEAD  
Reverting HEAD+ 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
> git tag v1.0.0-b1 -am 'Tag v1.0.0-b1'
> sqitch bundle --tags-only
Bundling in bundle/
Config written to bundle/sqitch.conf
Bundling v1.0.0-dev1
- appuser
- users
- change_pass
- insert_user
Bundling v1.0.0-b1
- change_pass_v2
- insert_user_v2
Plan written to bundle/sqitch.plan
>
```



What About Bundling?

```
Terminal  
> git tag v1.0.0-b1 -am 'Tag v1.0.0-b1'  
> sqitch bundle --tags-only  
Bundling in bundle/  
Config written to bundle/sqitch.conf  
Bundling v1.0.0-dev1  
- appuser  
- users  
- change_pass  
- insert_user  
Bundling v1.0.0-b1  
- change_pass_v2  
- insert_user_v2  
Plan written to bundle/sqitch.plan  
>
```



What About Bundling?

```
Terminal  
> git tag v1.0.0-b1 -am 'Tag v1.0.0-b1'  
> sqitch bundle --tags-only  
Bundling in bundle/  
Config written to bundle/sqitch.conf  
Bundling v1.0.0-dev1  
- appuser  
- users  
- change_pass  
- insert_user  
Bundling v1.0.0-b1  
- change_pass_v2  
- insert_user_v2  
Plan written to bundle/sqitch.plan  
>
```

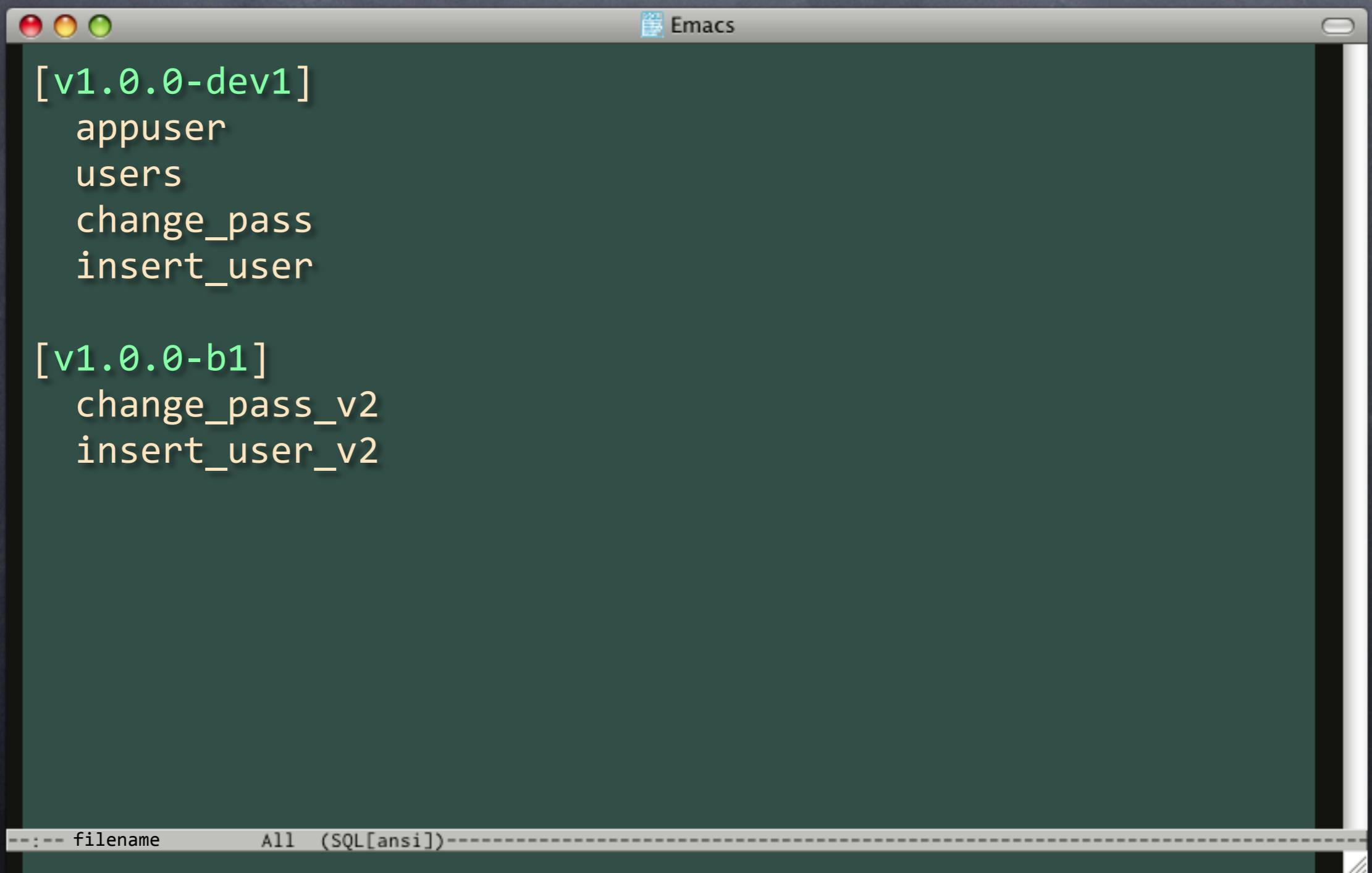


What About Bundling?

```
Terminal  
> git tag v1.0.0-b1 -am 'Tag v1.0.0-b1'  
> sqitch bundle --tags-only  
Bundling in bundle/  
Config written to bundle/sqitch.conf  
Bundling v1.0.0-dev1  
- appuser  
- users  
- change_pass  
- insert_user  
Bundling v1.0.0-b1  
- change_pass_v2  
- insert_user_v2  
Plan written to bundle/sqitch.plan  
> emacs bundle/sqitch.plan
```



What's the Plan?

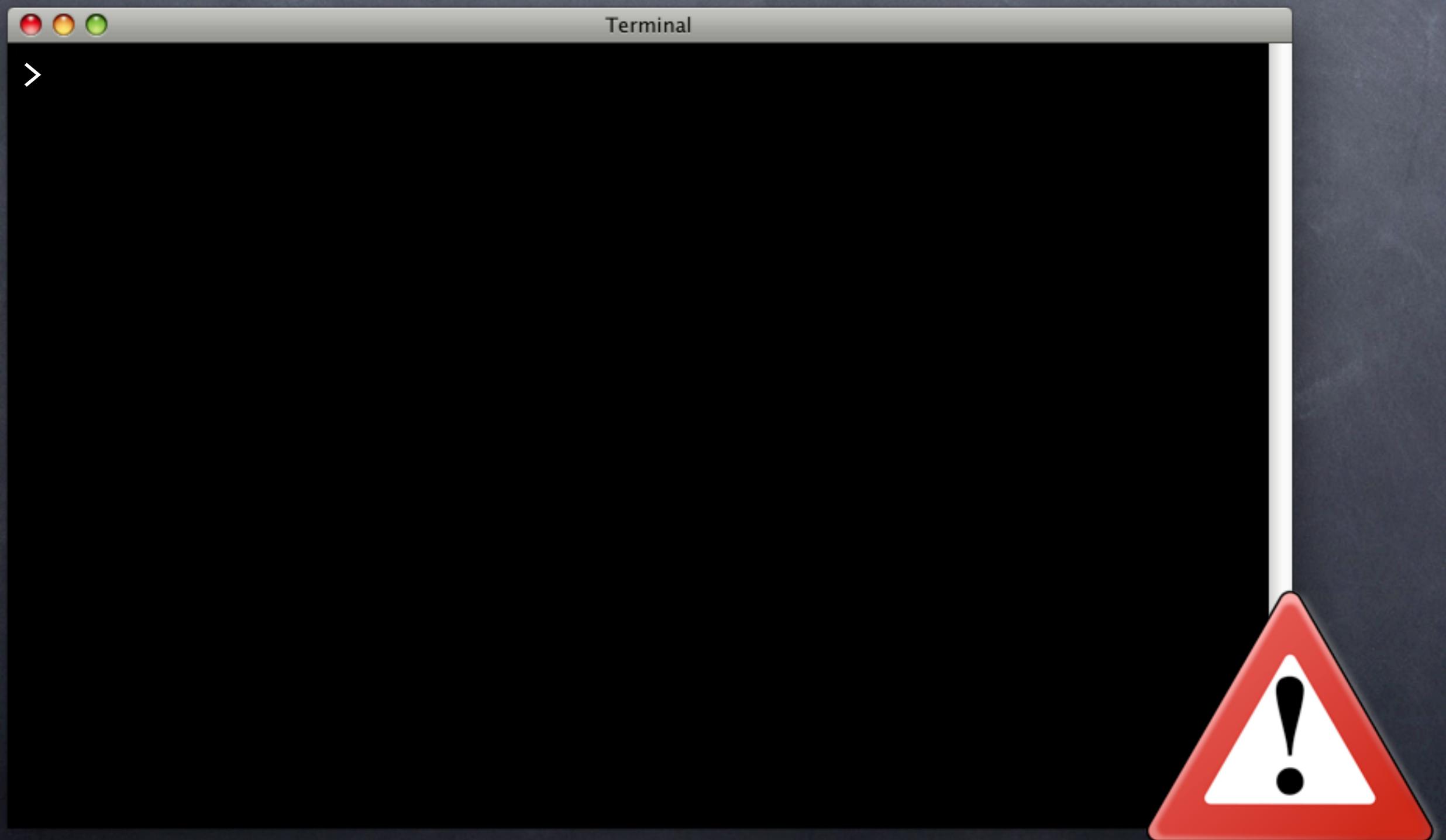


The image shows a screenshot of an Emacs window with a dark green background. The window title is "Emacs". The content is organized into two main sections, each starting with a green bracketed identifier:

- [v1.0.0-dev1]
 - appuser
 - users
 - change_pass
 - insert_user
- [v1.0.0-b1]
 - change_pass_v2
 - insert_user_v2

At the bottom of the window, there is a status bar with the following text:
--- filename All (SQL[ansi]) ---

Make it So



Make it So

```
Terminal  
> cd bundle  
> sqitch -d flipr_staging deploy  
Deploying v1.0.0-b1 to flipr_staging  
+ change_pass_v2  
+ insert_user_v2  
>
```



Make it So

```
Terminal  
> cd bundle  
> sqitch -d flipr_staging deploy  
Deploying v1.0.0-b1 to flipr_staging  
+ change_pass_v2  
+ insert_user_v2  
> sqitch -d flipr_staging status  
# On database flipr_staging  
# Tag: v1.0.0-b1  
# Step: insert_user_v2  
# Date: 2012-05-18 17:51:35  
#  
Nothing to deploy (up-to-date)  
>
```



Ship It!

Other Commands



Other Commands

- log - Like git log



Other Commands

- `log` - Like git log
- `check` - Validate plan & database



Other Commands

- `log` - Like git log
- `check` - Validate plan & database
- `test` - Test deployment success



Other Commands

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



Current Status



Current Status

- ➊ Working on it full time



Current Status

- Working on it full time
- Rethinking VCS integration



Current Status

- Working on it full time
- Rethinking VCS integration
- Rely more on plan file



Current Status

- Working on it full time
- Rethinking VCS integration
 - Rely more on plan file
 - Complement with VCS history



Current Status

- Working on it full time
- Rethinking VCS integration
 - Rely more on plan file
 - Complement with VCS history
- Tools for plan file management



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.

Simple SQL Change Management with Sqitch

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

PGCon 2012
Ottawa, Canada