

# Rails Bootcamp

March 14, 2010



Jeremy Lightsmith  
Wes Maldonado

# What's Going On?

# Goals

# Exposure

# Repetition

# Speed

**As much as you can take**

# Not Enough Time



# Accelerators

# Show, Tell, & Do

# Pair Programming

one person types?

# Stories

Always  
Be  
Cheating

**FULL**

# The Day

10a-11:20a	Part 1: Blog
11:30a-1p	Part 2: Quote of the Day App
1p-2p	Lunch
2p-3:20p	Part 3: Task List App
3:30p-5p	Part 4: Music Database



# The Parking Lot

# Part I : Blog

# Links

[http://github.com/jeremylightsmith/  
rails\\_bootcamp](http://github.com/jeremylightsmith/rails_bootcamp)

<http://guides.rubyonrails.org/>

<http://apidock.com/>

# Pass the Stewie

Model  
View  
Controller

# STORY I-I

# Directory Structure

File/Folder	Purpose
README	This is a brief instruction manual for your application. Use it to tell others what your application does, how to set it up, and so on.
Rakefile	This file contains batch jobs that can be run from the terminal.
app/	Contains the controllers, models, and views for your application. You'll focus on this folder for the remainder of this guide.
config/	Configure your application's runtime rules, routes, database, and more.
db/	Shows your current database schema, as well as the database migrations. You'll learn about migrations shortly.
doc/	In-depth documentation for your application.
lib/	Extended modules for your application (not covered in this guide).
log/	Application log files.
public/	The only folder seen to the world as-is. This is where your images, javascript, stylesheets (CSS), and other static files go.
script/	Scripts provided by Rails to do recurring tasks, such as benchmarking, plugin installation, and starting the console or the web server.
test/	Unit tests, fixtures, and other test apparatus. These are covered in <a href="#">Testing Rails Applications</a>
tmp/	Temporary files
vendor/	A place for third-party code. In a typical Rails application, this includes Ruby Gems, the Rails source code (if you install it into your project) and plugins containing additional prepackaged functionality.

[http://guides.rubyonrails.org/getting\\_started.html](http://guides.rubyonrails.org/getting_started.html)



# Actions

CRUD	Rails method
create	(new) create
read	show, index
update	(edit) update
delete	destroy

# Naming Conventions

- partials start w/ “\_”

# STORY 1-2, 1-3

## Models

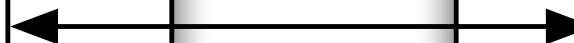
Post
name
content

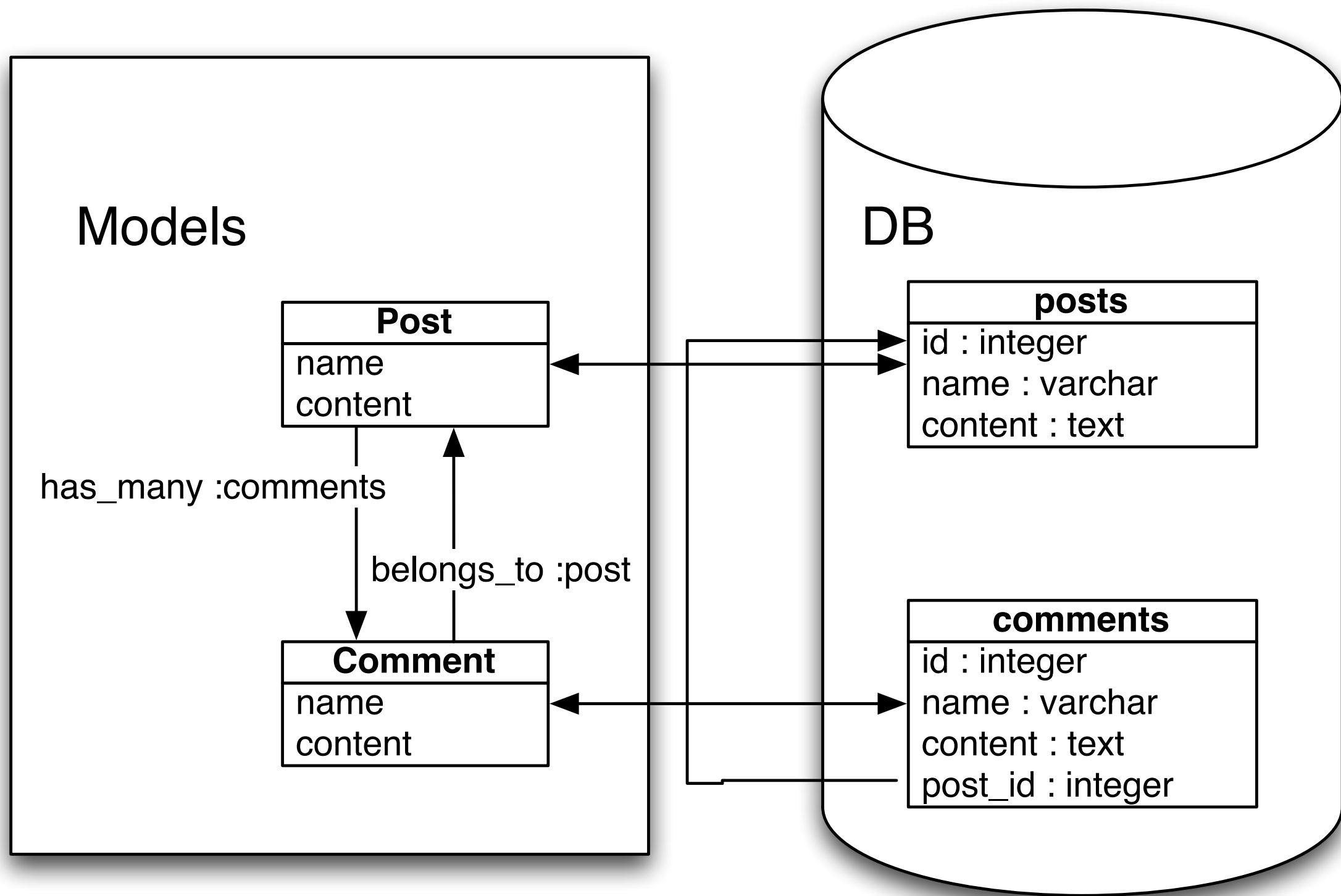
Comment
name
content

## DB

posts
id : integer
name : varchar
content : text

comments
id : integer
name : varchar
content : text





**Let's see it**

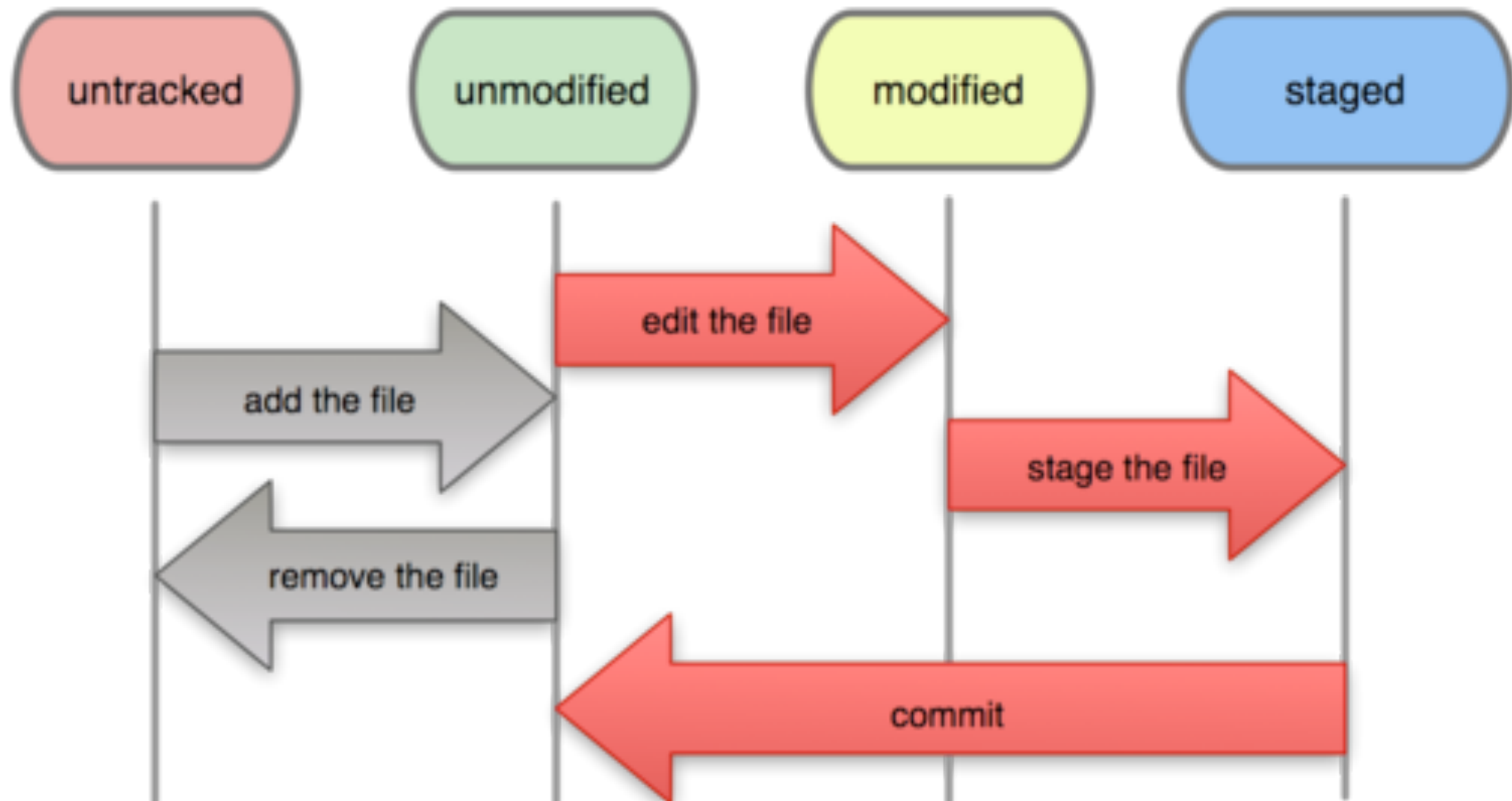


# Part 2 :

# Quote of the Day App

# GIT

# File Status Lifecycle

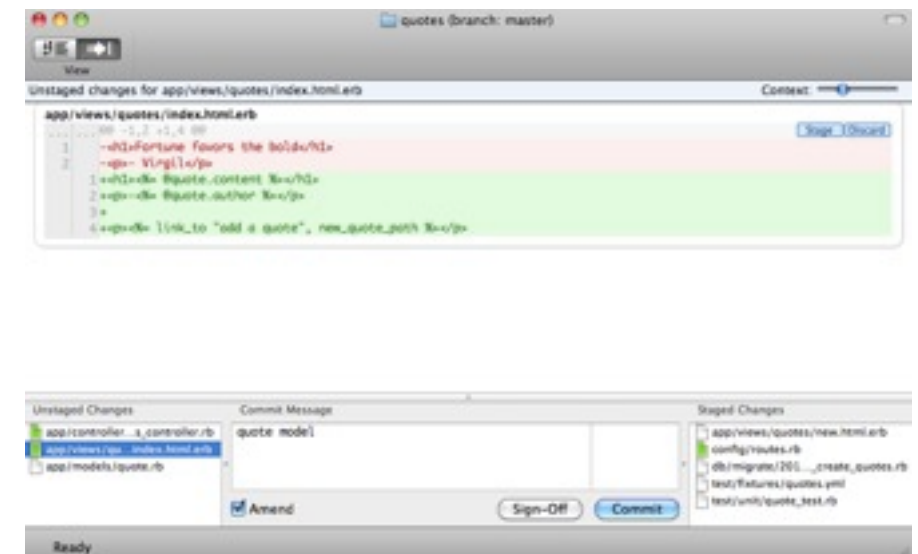
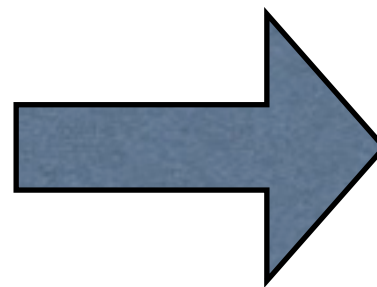


<http://progit.org/book/ch2-2.html>

git push [remote] [branch]  
git pull [remote] [branch]

git add .  
git commit -m "..."  
git status  
git log

or

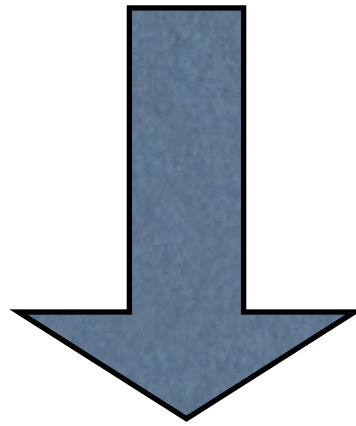


# Heroku

<http://heroku.com/>

# Routes

```
ActionController::Routing::Routes.draw do |map|  
  map.root :controller => "quotes", :action => "index"  
  map.resources :quotes  
end
```



**redirect\_to** root\_path

new\_quote\_path  
quotes\_path  
edit\_quotes\_path  
quote\_path(quote)

# LUNCH



# Part 3 :

# A Task List App

# Migrations

# REST

every **RESOURCE**  
has **ONE** name

consistent **VERBS**  
to access **RESOURCES**

**GET**  
**PUT** **POST**  
**DELETE**

CRUD	REST verb	Rails method
create	POST	create (new)
read	GET	show, index
update	PUT	update (edit)
delete	DELETE	destroy

# Part 4 :

# A Music Database

# Nested Resources



# Images

# Uploading

# Resizing

# On the cloud (Heroku, etc...)

# Other stuff

# Testing

Do it

# Test-Driven Development



# Troubleshooting

# Check the server console

# Check the log

# CSS

# Javascript

# E-mail

# Logging In

# Use a plugin :

authlogic is a good choice  
(there are a bunch out there)