

MODELS, SCAFFOLDING AND MIGRATIONS

A MINI LECTURE



CHALLENGES

Rebecca challenged you to shower together, save water

Accept

Decline

What's Happening



Victoria completed 3 goals and earned 60 points. 8 minutes ago
↳ Buy a toy made from renewable materials
↳ Turn thermostat down by 4 degrees in cold weather
↳ Watch Annie Leonard's online video, The Story of Stuff



Robin earned the **Conscious Consumer** badge! 31 minutes ago



Robin completed 7 goals and earned 150 points. 39 minutes ago
↳ Recycle printer cartridges
↳ Burn natural candles
↳ Use cruise control as much as possible
...and 4 more



Barbara completed 5 goals and earned 175 points. about an hour ago
↳ Eliminate styrofoam, especially from food and drink purchases
↳ Purchase or lease a hybrid
↳ Pick up litter and trash when you see it
...and 2 more



Cathy completed 4 goals and earned 70 points. about an hour ago
↳ Use eco-friendly office supplies
↳ Use non-toxic green cleaning products in your work area or office
↳ Switch to soy-based inks and remanufactured toner cartridges
...and 1 more



Michaela completed a goal and earned 20 points. about an hour ago

Ideas for You

NEW!

Walk or bike to do your regular errands
Mark Complete

Carpool to school one day a week
Mark Complete

Keep a stash of plastic shopping bags in
the office kitchen
Mark Complete

Your Upcoming Goals

THIS WEEK

Switch to natural hair gel

Buy organic coffee regularly

LATER

Switch to a natural or organic mattress

View Your Goals



≡ Invite Friends ≡

Email

Email

Email



I KNOW THAT FEEL SIS



SCAFFOLDING

Rails scaffolding is a quick way to generate some of the major pieces of an application.





WTF?

```
rails generate scaffold Post name:string title:string content:text
```

```
invoke active_record
create db/migrate/20120816011924_create_posts.rb
create app/models/post.rb
invoke rspec
create spec/models/post_spec.rb
invoke resource_route
route resources :posts
invoke scaffold_controller
create app/controllers/posts_controller.rb
invoke erb
create app/views/posts
create app/views/posts/index.html.erb
create app/views/posts/edit.html.erb
create app/views/posts/show.html.erb
create app/views/posts/new.html.erb
create app/views/posts/_form.html.erb
invoke rspec
create spec/controllers/posts_controller_spec.rb
create spec/views/posts/edit.html.erb_spec.rb
create spec/views/posts/index.html.erb_spec.rb
create spec/views/posts/new.html.erb_spec.rb
create spec/views/posts/show.html.erb_spec.rb
create spec/routing/posts_routing_spec.rb
invoke rspec
create spec/requests/posts_spec.rb
invoke helper
create app/helpers/posts_helper.rb
invoke rspec
create spec/helpers/posts_helper_spec.rb
invoke assets
invoke coffee
create app/assets/javascripts/posts.js.coffee
invoke scss
create app/assets/stylesheets/posts.css.scss
invoke scss
create app/assets/stylesheets/scaffolds.css.scss
```

Listing posts

Name Title Content

New Post

Editing post

Name

Liana

Title

ror is awesome

Content

zomg, I can't believe how easy this is!

Update Post

[Show](#) | [Back](#)

Post was successfully created.

Post was successfully created.

Name: Liana

Title: ror is awesome

Content: zomg, I can't believe how easy this is!

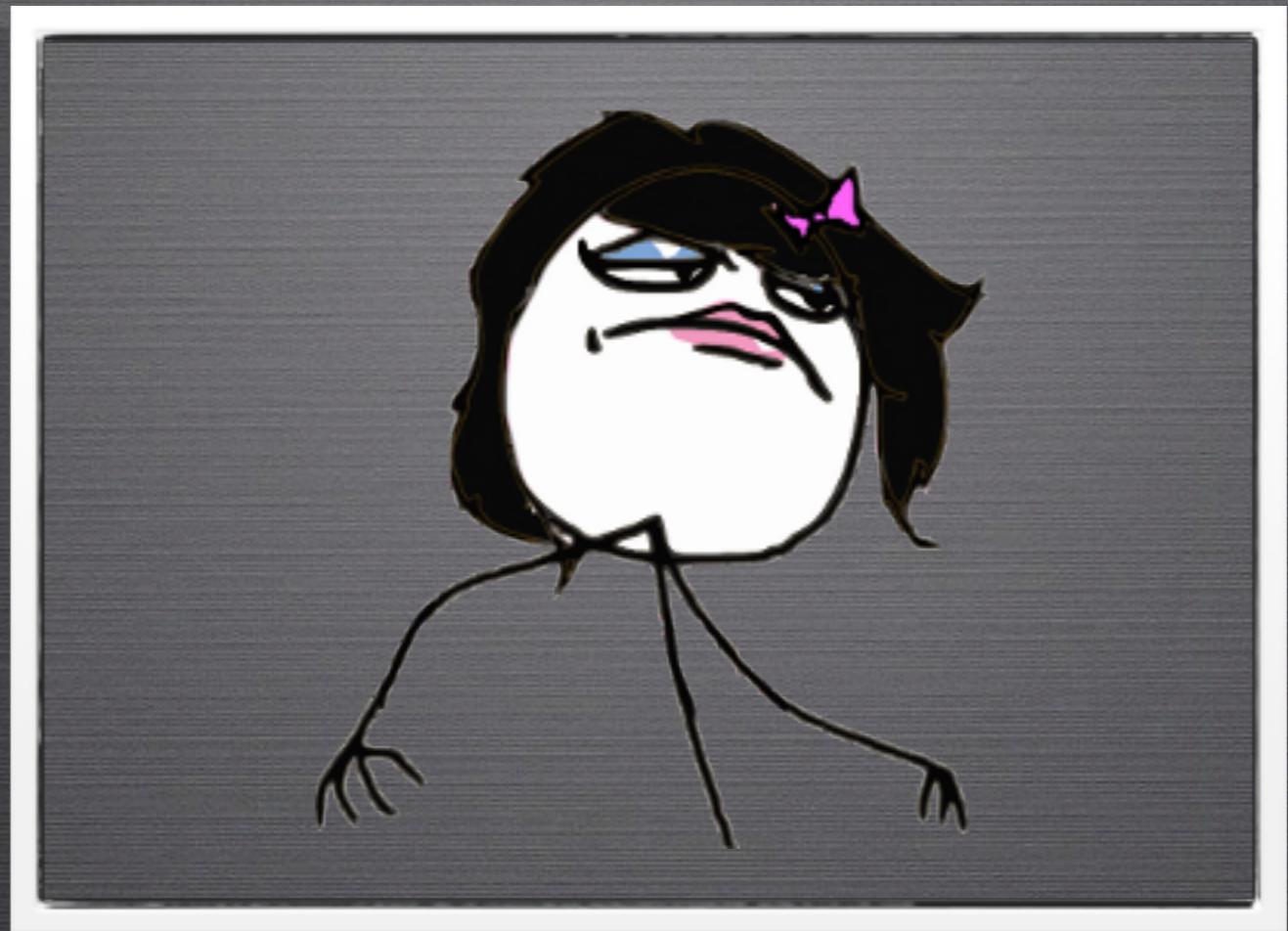
[Edit](#) | [Back](#)

Listing posts

Name Title Content

Liana ror is awesomezomg, I can't believe how easy this is! [Show](#) [Edit](#) [Destroy](#)

[New Post](#)



AWWW, YEAH

MIGRATIONS

Migrations are Ruby classes that make it simple to create and modify database tables.



But I already know SQL...

```
rake db:migrate
```

```
class CreatePosts < ActiveRecord::Migration
  def up
    create_table :posts do |t|
      t.string :name
      t.string :title
      t.text :content

      t.timestamps
    end
  end

  def down
    drop_table :posts
  end
end
```

```
rake db:rollback
```

- add_column
- add_index
- change_column
- change_table
- create_table
- drop_table
- remove_column
- remove_index
- rename_column
- execute



MODELS

VALIDATIONS

Model-level validations ensure that only valid data is saved into your database.



Why can't I just have some validation...

```
class User < ActiveRecord::Base

  validates :name, :presence=>true, :length=> { :minimum => 15 }

  validates :title, :allow_blank => true

  validates :email, :presence=>true, :uniqueness => true

  validates :bio, :length => { :maximum => 500 }

  validates :age, :numericality => true

  validate :valid_vocal_range

  def valid_vocal_range
    if high_note < low_note
      errors.add(:high_note, "this is not a valid vocal range")
    end
  end

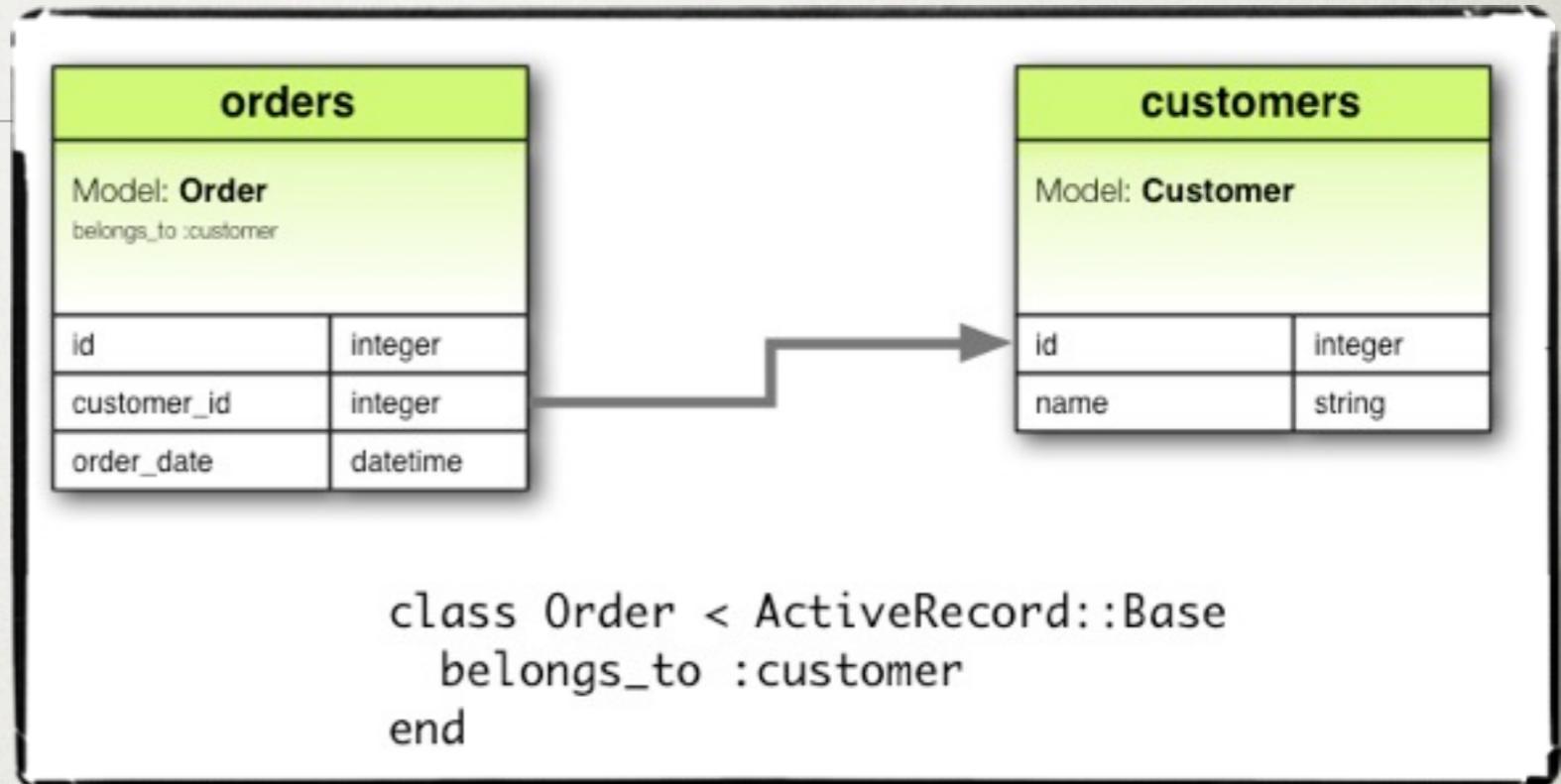
end
```

ASSOCIATIONS

Active Record associations tells Rails that there is a connection between two models.



I never knew it could be like this...



- `belongs_to`
- `has_one`
- `has_many`
- `has_many :through`
- `has_and_belongs_to_many`

```
class Customer < ActiveRecord::Base
  has_many :orders, :dependent => :destroy
end

class Order < ActiveRecord::Base
  belongs_to :customer
end

@order = @customer.orders.create(:order_date=>Time.now)

Customer.destroy(@customer)
```

WANNA LEARN MORE?



RESOURCES

- Rails Guides: <http://guides.rubyonrails.org>
- RailsCasts by Ryan Bates: [http:// railscasts.com](http://railscasts.com)
- DevChix: <http:// www.devchix.com>
- [http:// groups.google.com/group/ror-studygroup-for-women-boston](http://groups.google.com/group/ror-studygroup-for-women-boston)
- [http:// groups.google.com/group/nyc-ruby-woman](http://groups.google.com/group/nyc-ruby-woman)
- [http:// groups.google.com/group/boston-rubygroup](http://groups.google.com/group/boston-rubygroup)



THANKS!

SEE YOU AT THE NEXT
BOSTON.RB MEETUP!



LIANA LEAHY
TWITTER: @LLEAHY