

Lab 6 – Warm Up



2 Associations between models

Associations

3

- `belong_to`
 - ▣ sets up a one-to-one connection with another model, such that each instance of the declaring model “belongs to” one instance of the other model
- `has_one`
 - ▣ sets up a one-to-one connection with another model, such that each instance of a model contains or possesses one instance of another model
- `has_many`
 - ▣ one-to-many connection between models
 - ▣ typically on the other side of a `belong_to` association

Associations

4

- `has_many`, `has_one :through`
 - ▣ **similar to** `has_many` **and** `has_one`
 - ▣ **but the declaring model can be matched with one instance of another model by proceeding *through* a third model**
- `has_and_belongs_to_many`
 - ▣ **creates a direct many-to-many connection with another model, with no intervening model**
- **For each of them, Rails generates some methods**

In our case...

5

- 1 post belongs to 1 user
 - ▣ in the Post model we'll have: `belongs_to :user`

Post		
id	content	user_id
1	Some text...	1

Post.user

User		
id	name	email
1	Luigi	...

belongs_to



In our case...

6

- 1 user can have many post
 - ▣ in the User model we'll have: `has_many :posts`

User.posts

Post		
id	content	user_id
1	Some text...	1

Post		
id	content	user_id
50	Other text...	1

User		
id	name	email
1	Luigi	...

has_many

has_many

In our case...

7

belongs_to	<code>post.user</code>	return the User object associated with the post
has_many	<code>user.posts</code>	return an Array of the user's posts
	<code>user.posts.create(arg)</code>	create a post (<code>user_id = user.id</code>)
	<code>user.posts.create!(arg)</code>	create a post, exception on failure
	<code>user.posts.build(arg)</code>	return a new Post object (<code>user_id = user.id</code>)

8

Some tips...

default_scope

9

- Use this macro in your model to set a default scope for all operations on the model
- Example

```
class Post < ActiveRecord::Base
  # descending order for getting the posts
  default_scope order: 'posts.created_at
    DESC `
end
```

```
Post.all #=> SELECT * FROM posts ORDER BY
  created_at DESC
```

Generate fake posts

10

□ With the Faker gem

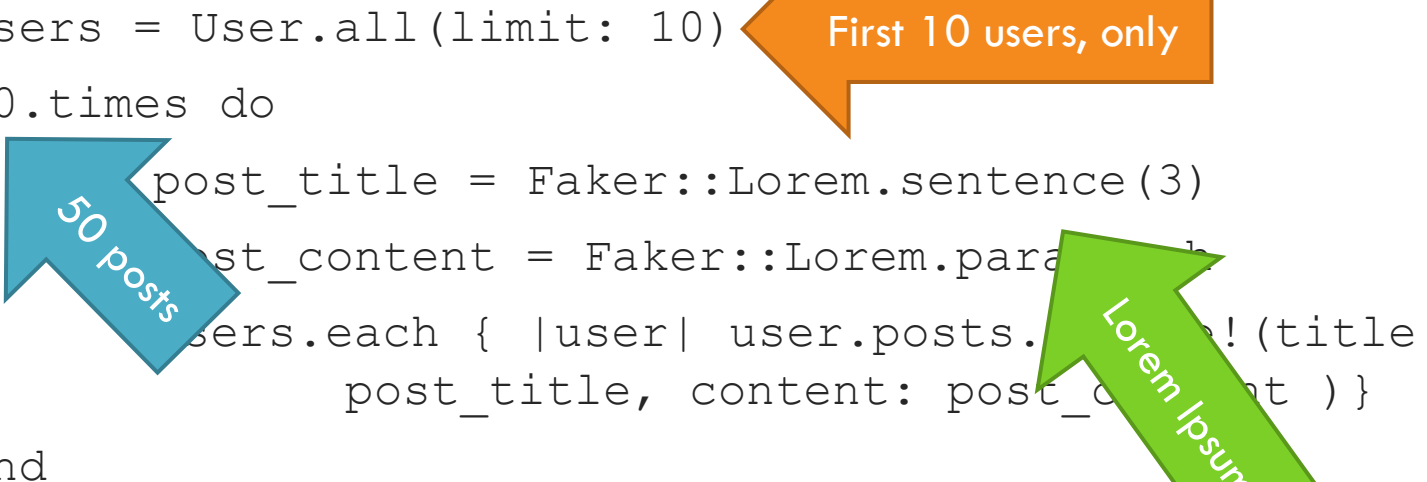
```
users = User.all(limit: 10)
50.times do
  post_title = Faker::Lorem.sentence(3)
  post_content = Faker::Lorem.paragraph
  users.each { |user| user.posts.create!(title:
    post_title, content: post_content )}
end
```

Generate fake posts

11

□ With the Faker gem




```
users = User.all(limit: 10)
50.times do
  post_title = Faker::Lorem.sentence(3)
  post_content = Faker::Lorem.paragraph
  users.each { |user| user.posts.create!(title:
    post_title, content: post_content ) }
end
```



License



12

- This work is licensed under the Creative Commons “Attribution-NonCommercial-ShareAlike Unported (CC BY-NC-SA 3,0)” License.
- You are free:
 - **to Share** - to copy, distribute and transmit the work
 - **to Remix** - to adapt the work
- Under the following conditions:
 - **Attribution** - You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). 
 - **Noncommercial** - You may not use this work for commercial purposes. 
 - **Share Alike** - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one. 
- To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>