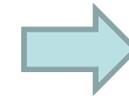
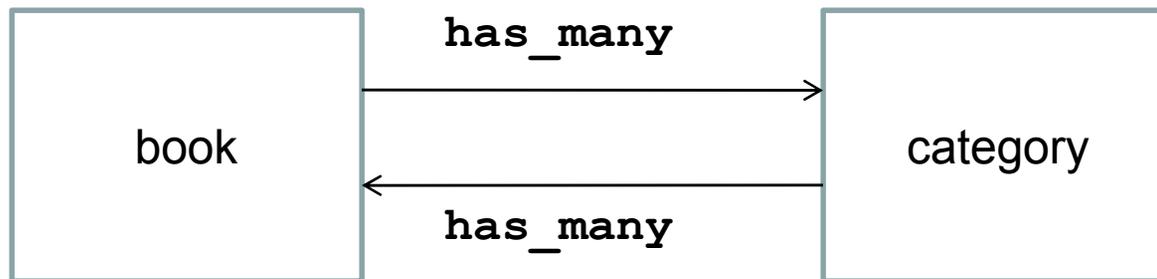


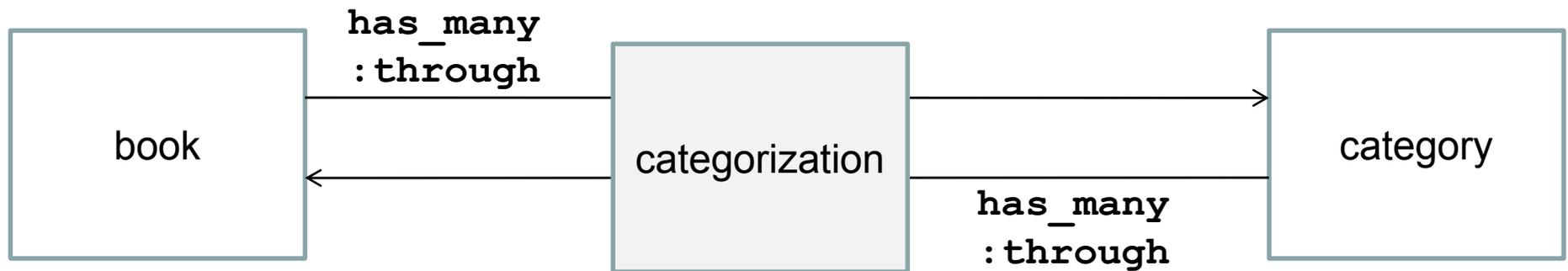
Relationships between tables

Many-to-many



Relationships between tables

Many-to-many



Viewing tables with Sqliteman

books

	id	title	isbn	creat
1	1	Ruby on Rails For Dummies	12233355566622444	2008-01-22
2	2	Java For Dummies	111222333	2008-01-22

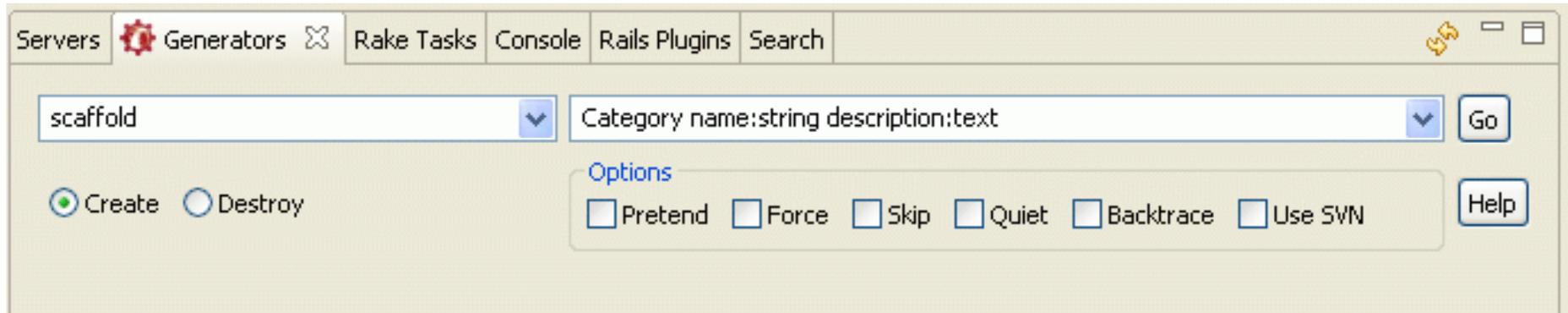
categorizations

	id	book_id	category_id	created_
1	25	1	2	2008-02-03 21
2	26	1	3	2008-02-03 21

categories

	id	name	description	
1	1	Computers	Computer Science, Information Systems, Computer Engineering	2008
2	2	English	Composition, Creative Writing, Literature	2008
3	3	Social Sciences	Psychology, Sociology, Anthropology	2008

Create categories



The screenshot shows the Rails Generators interface. At the top, there are tabs for Servers, Generators (active), Rake Tasks, Console, Rails Plugins, and Search. Below the tabs, there is a dropdown menu with 'scaffold' selected. To the right of this is a text input field containing 'Category name:string description:text'. Below the dropdown and text input are two radio buttons: 'Create' (selected) and 'Destroy'. To the right of the radio buttons is an 'Options' section with several checkboxes: Pretend, Force, Skip, Quiet, Backtrace, and Use SVN. A 'Go' button is located to the right of the text input field, and a 'Help' button is located to the right of the Options section.

Servers  Generators  Rake Tasks Console Rails Plugins Search

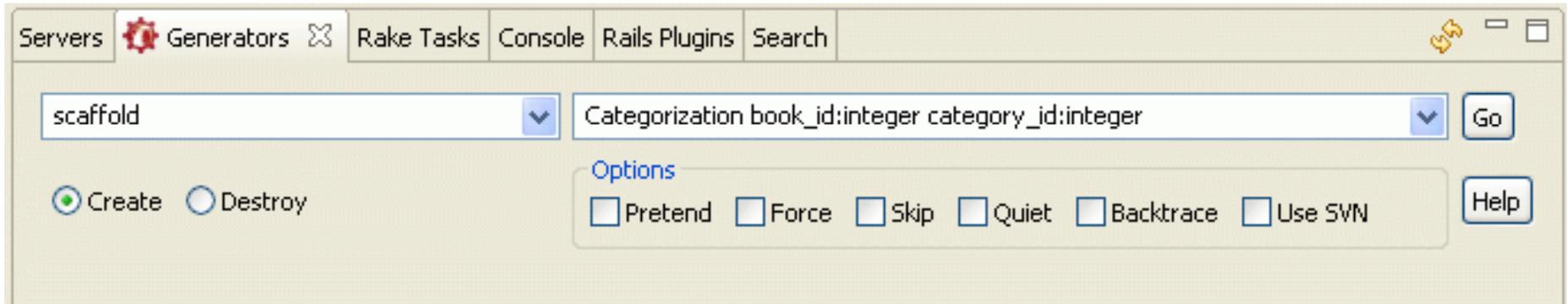
scaffold Category name:string description:text Go

Create Destroy

Options

Pretend Force Skip Quiet Backtrace Use SVN Help

Create categorizations



Servers  Generators  Rake Tasks Console Rails Plugins Search   

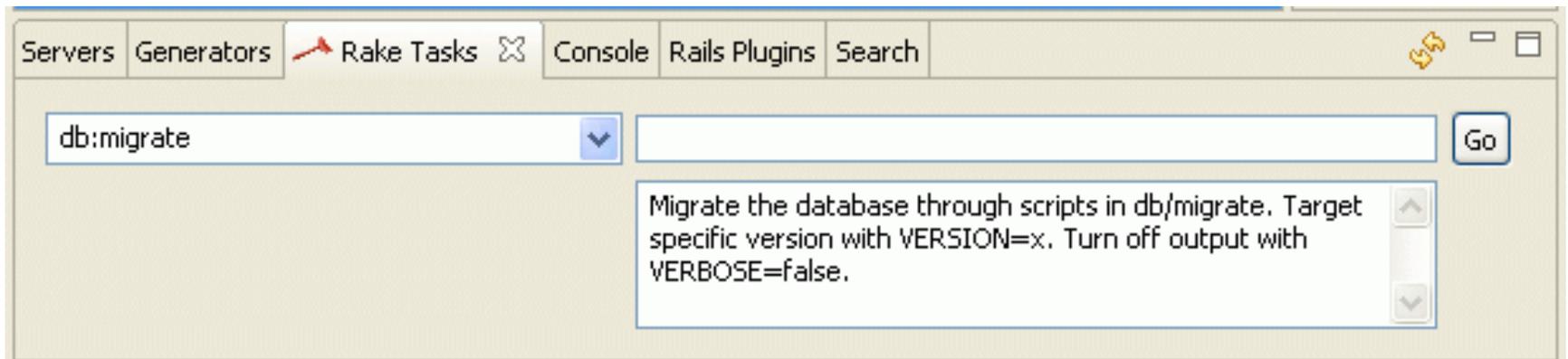
scaffold  Categorization book_id:integer category_id:integer 

Create Destroy

Options

Pretend Force Skip Quiet Backtrace Use SVN

Migrate both at once



```
book.rb x
class Book < ActiveRecord::Base
  validates_presence_of :title
  validates_inclusion_of :isbn, :in => 11111..999999999

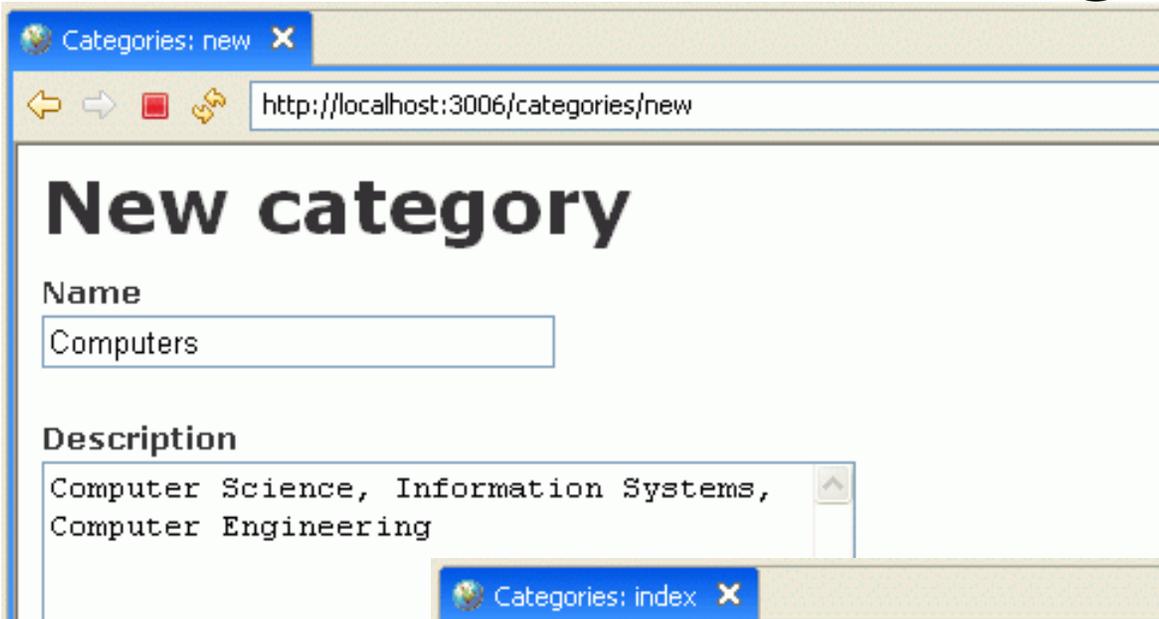
  has_many :reviews

  has_many :categorizations
  has_many :categories, through => :categorizations
end
```

```
categorization.rb x
class Categorization < ActiveRecord::Base
  belongs_to :book
  belongs_to :category
end
```

```
category.rb x
class Category < ActiveRecord::Base
  has_many :categorizations
  has_many :books, through => :categorizations
end
```

Create categories



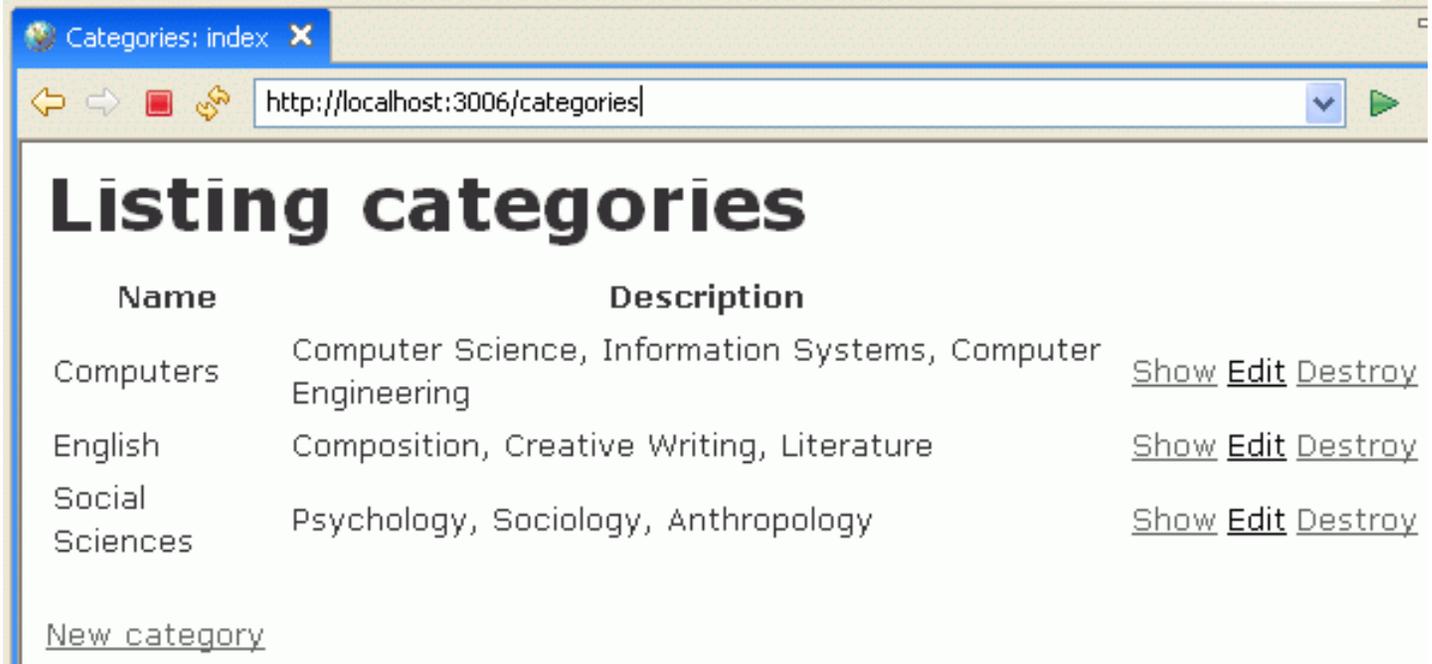
Categories: new x

http://localhost:3006/categories/new

New category

Name

Description



Categories: index x

http://localhost:3006/categories/

Listing categories

Name	Description	
Computers	Computer Science, Information Systems, Computer Engineering	Show Edit Destroy
English	Composition, Creative Writing, Literature	Show Edit Destroy
Social Sciences	Psychology, Sociology, Anthropology	Show Edit Destroy

[New category](#)

Add a method to

Bookstore\app\controllers\books_controller.rb

```
books_controller.rb x
1 class BooksController < ApplicationController
2
3   def set_categories
4     @book = Book.find(params[:id])
5     for category in Category.find(:all)
6       @book.categories.delete(category)
7       if params['category'+category.id.to_s]['isChecked'] == '1'
8         @book.categories << category
9       end
10    end
11    redirect_to :action => 'show', :id => params[:id]
12  end
13
14  def add_review
15    Book.find(params[:id])
```

Add code to

Bookstore\app\views\books\show.html.erb

```
show.html.erb x
8   <%=h @book.isbn %>
9   </p>
10
11  <% form_tag :action => "set_categories", :id => @book do %>
12    <% @categories = Category.find(:all) %>
13    <b>Categories</b><br />
14    <% @categories.each { |category|
15      isChecked = @book.categories.include?category %>
16      <%= check_box('category'+category.id.to_s, 'isItChecked',
17        { :checked => ((isChecked)?"checked":"unchecked") } ) %>
18      <%= ((isChecked)?"<b>":"") + category.name +
19        ((isChecked)?"</b>":"" ) %><br />
20    <% } %>
21    <%= submit_tag %>
22  <% end %>
23
24  <p>
25  <b>Reviews:</b><br />
```

