



devise gem

Add devise to your Gemfile

```
gem 'devise'
```

and run **bundle install**

Run the Devise installer

```
$ rails generate devise:install
```

Follow the CLI instructions

- Setup a root path
- Add default_url_options to development env
- Add flash notice/alerts
- Run the command **rails g devise:views**

Restart your server

Create your user model using devise

```
$ rails generate devise user
```

```
$ rake db:migrate
```

```
$ rake routes
```

Visit your sign up page

http://localhost:3000/users/sign_up

Scaffolding

Create our model and controller

```
$ rails generate model pin description:string
```

```
$ rails generate controller pins
```

We can use a scaffold instead

\$ rails generate scaffold pin

warning!

Associations

Let's add a user_id to the pin model

```
$ rails generate migration  
add_user_id_to_pins user_id:integer:index
```

```
$ rake db:migrate
```

Creating an order for a customer

```
class Customer < ActiveRecord::Base  
end
```

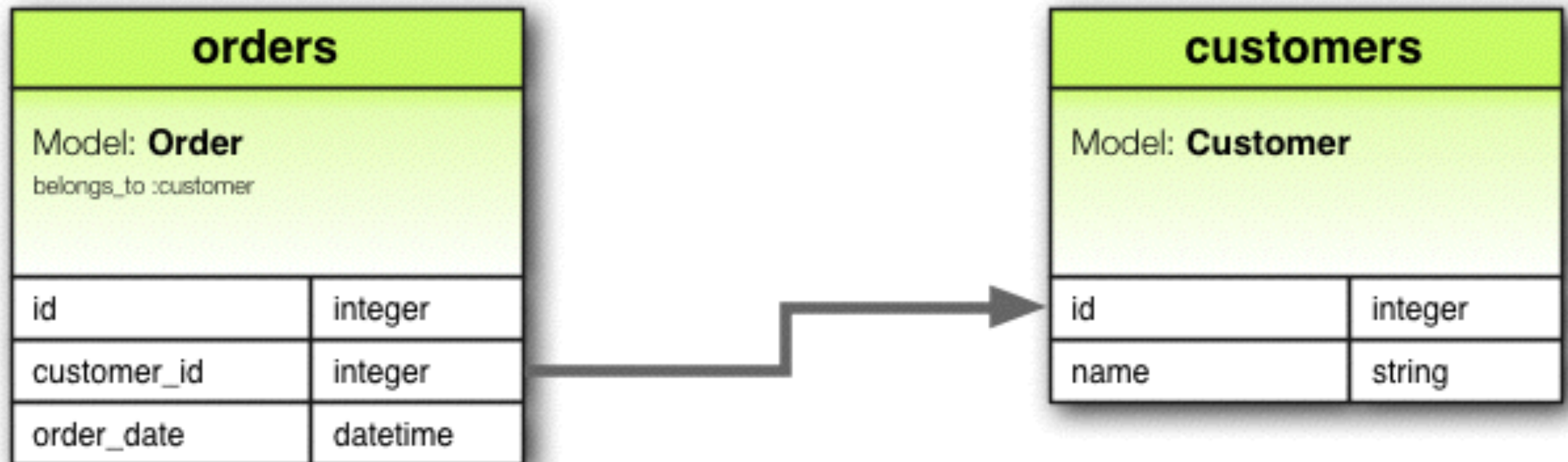
```
class Order < ActiveRecord::Base  
end
```

```
@order = Order.create(  
  order_date: Time.now,  
  customer_id: @customer.id  
)
```

Active Record Associations

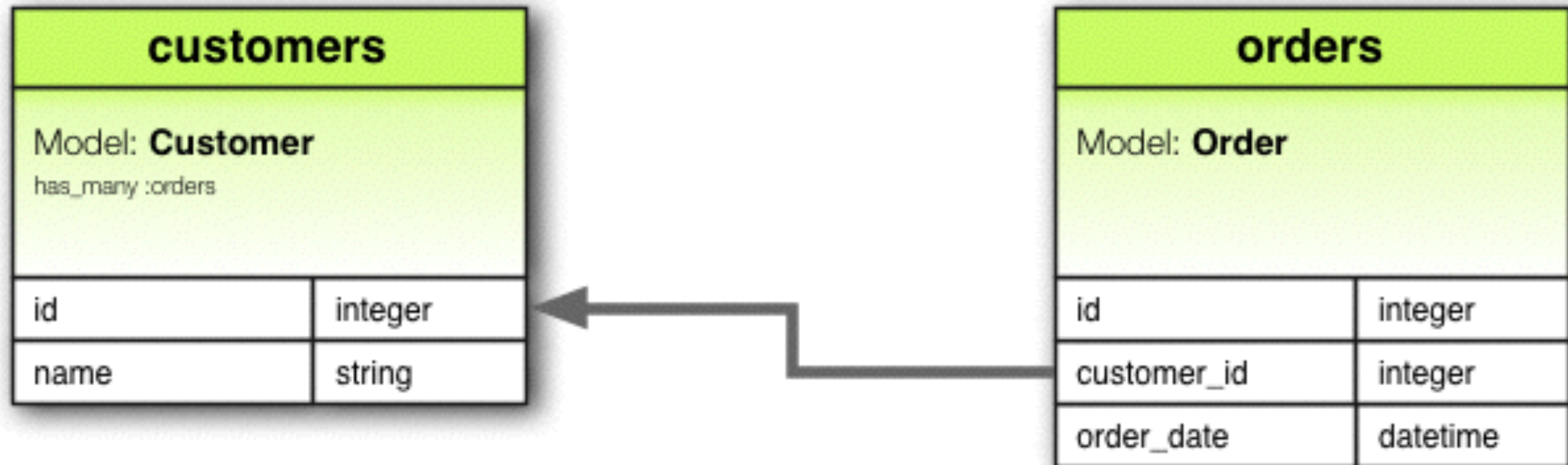
- belongs_to
- has_one
- has_many
- has_many :through
- has_one :through
- has_and_belongs_to_many

belongs_to



```
class Order < ActiveRecord::Base
  belongs_to :customer
end
```


has_many



```
class Customer < ActiveRecord::Base
  has_many :orders
end
```

Creating an order for a customer

```
class Customer < ActiveRecord::Base
  has_many :orders
end
```

```
class Order < ActiveRecord::Base
  belongs_to :customer
end
```

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

Create an empty order for a customer

```
@customer = Customer.find(1)  
@new_order = @customer.orders.build
```

versus

```
@new_order = Order.new(customer_id : 1)
```

Let's update our
PinsController

def new

```
def new
  @pin = current_user.pins.build
end
```

def create

```
def create
  @pin = current_user.pins.build(pin_params)
  if @pin.save
    redirect_to @pin
  else
    render 'new'
  end
end
```

Locking down the pins

/app/controllers/pins_controller.rb

```
before_action :authenticate_user!, except:  
[:index, :show]
```

```
before_action :correct_user, only:  
[:edit, :update, :destroy]
```


/app/controllers/pins_controller.rb

```
private
def correct_user
  @pin = current_user.pins.find_by(id: params[:id])
  if @pin.nil?
    redirect_to pins_path,
      notice: "Not authorized to edit this pin"
  end
end
```

Pinterest Project

<http://github.com/rcacademy/pinteresting>