

# Rails Girls App Tutorial

## 4. Add another entity

### 4.1. Create a models scaffold

**Coach:** Add a new entity to the application. Go with scaffold or migrations.

Example: Create a scaffold, with the name, the body (contents of the model) and with the reference to the initial model table (example: `idea_id`).

```
rails generate scaffold <model_name> <attribute>:<type>
```

example:

```
rails g scaffold comment user_name:string body:text idea_id:integer
```

This will create a migration file that lets your database know about the new comments table. Run the migrations using

```
rails db:migrate
```

### 4.2. Add relations to models

You need to make sure that Rails knows the relation between objects (ideas and comments). As one idea can have many comments we need to make sure the idea model knows that. Open `app/models/<initial-model>.rb` and below the row

```
class <Your-initial-model> < ApplicationRecord
```

add

```
has_many :<new-entity-pluralized>
```

The comment also has to know that it belongs to an idea. So open `app/models/<new-model>.rb` and below

```
class <New-model> < ApplicationRecord

example: class Comment < ApplicationRecord
```

add the row

```
belongs_to :<initial-model>
```

## 4.3. Render the form and existing entries

Going further with the ideas, we can open `app/views/<initial-model-pluralized>/show.html.erb` and after the `image_tag`

```
<%= image_tag(<model-name>.picture_url, :width => 600) if <model-name>.picture.present? %>
```

add

```
<h3>Title of new model</h3>
<% <new-model-pluralized>.each do |<new-model>| %>
  <div>
    <strong><%= <new-model>.user_name %></strong>
    <br>
    <p><%= <new-model>.body %></p>
    <p><%= link_to 'Delete', <new-model>_path(<new-model>), method: :delete, data: { confirm:
'Are you sure?' } %></p>
  </div>
<% end %>
<%= render partial: '<new-model-pluralized>/form', locals: { <new-model>: <new-model> } %>
```

In `app/controllers/<initial-model-pluralized>_controller.rb` add to the show action

```
@<new-model-pluralized> = @<initial-model>.<new-model-pluralized>.all
@<new-model> = @<initial-model>.<new-model-pluralized>.build

example:

@comments = @idea.comments.all
@comment = @idea.comments.build
```

Open `app/views/<new-model-pluralized>/_form.html.erb` and after

```
<div class="field">
  <%= form.label :body %><br>
  <%= form.text_area :body %>
</div>
```

add the row

```
<%= form.hidden_field :<model>_id %>
```

next, remove

```
<div class="field">
  <%= form.label :<model>_id %><br>
  <%= form.number_field :<model>_id %>
</div>
```

That's it. Now view an idea you have inserted to your application and there you should see the form for inserting a comment as well as deleting older comments.