

Model Relationships in Ruby on Rails

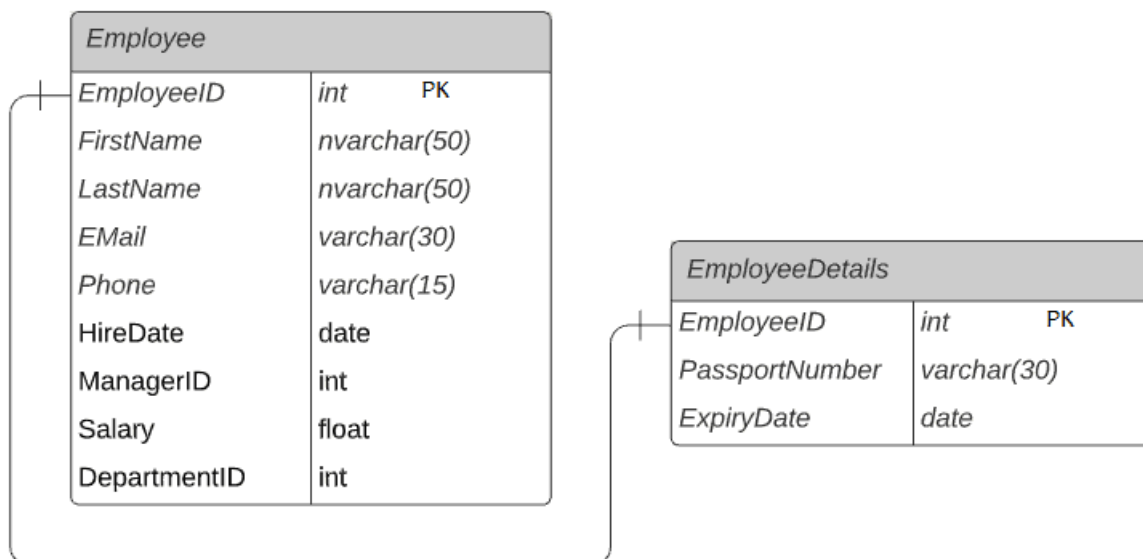
Table of Contents:

1. [_One-to-One_Relationship](#)
2. [_One-to-Many_Relationship](#)

In the Ruby on Rails framework, the relationship between models is often made using Active Record associations. This allows streamlined functionality that allows you to create something when one thing is created or delete something if the parent is deleted. The two associations that will be focused on in this documentation are One to One and One to Many Relationships.

One-to-One Relationship

In this type of relationship, there is a singular link from one model to another. For example, in our portfolio app we have a student who has one portfolio. In this relationship, the Primary student key is the foreign key in the portfolio table. Here is an example:



(Source of Image: <https://www.tutorialsteacher.com/sqlserver/tables-relations>)

This is just how we visualize it, here is how it is done in the actual code.

The association between Student and Portfolio is done in each model, in varying ways. In the student model, it will have a has_one association like so.

```
#https://guides.rubyonrails.org/v7.1/associations.html
has_one :portfolio, dependent: :destroy
```

Now in the portfolio model, you will see a belongs_to association being made. You can also see the dependent: :destroy feature here. This makes it so if the student gets destroyed, so does their portfolio. This helps keep your database clean!

```
class Portfolio < ApplicationRecord
  belongs_to :student
end
```

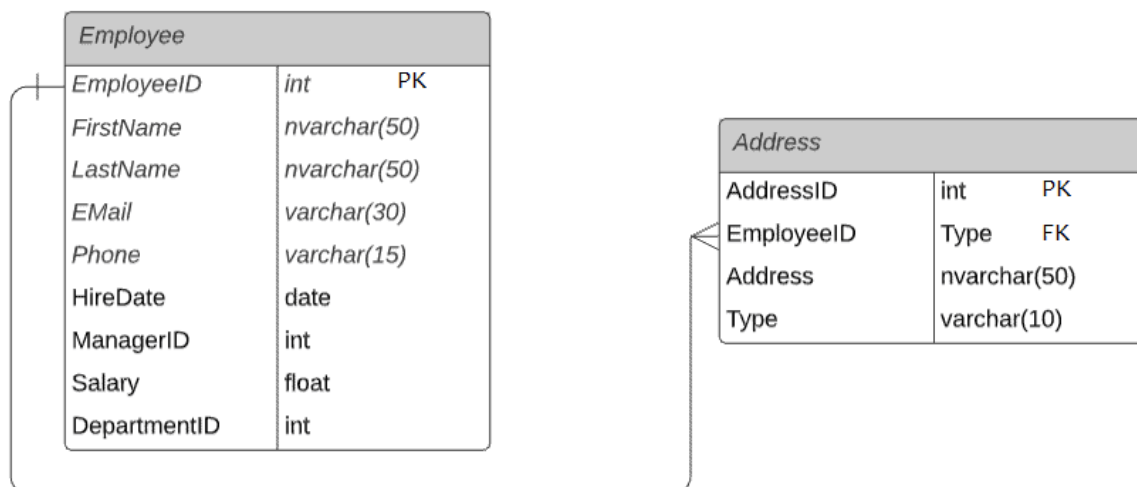
This makes the association between the two different models using Active Record.

You can scaffold this association by using the [student:references] command at the end of the scaffolding command.

Another important step is to make sure that your routes get nested correctly. In your routes.rb file, you can make it like this.

```
# Necessary Student routes and portfolio routes
resources :students, only: [:index, :show, :edit, :update, :destroy] do
  resource :portfolio, only: [:show, :edit, :update] # Nesting portfolio under student
end
```

One-to-Many Relationship



(Example of One-to-Many, same source as above model relationship)

The One-to-Many Relationship is quite similar to the One-to-One relationship. Instead of the `has_one` attachment, it instead has a `has_many` attachment in the Student model.

```
#has many relationship to projects  
has_many :projects , dependent: :destroy
```

But the project model still has the same `belongs_to` association that the Portfolio model had.

```
1 class Project < ApplicationRecord  
2   belongs_to :student  
3 end  
4
```

Like the One-to-One relationship, the Primary Key of the Student ID is the foreign key for the Project table. This allows the association to exist.

This information should allow you to create the relationships between models for any webapp you choose to work on!