

Fundamentals

Day 7

More ActiveRecord

Today's Objectives

- More ActiveRecord: Querying + Associations
- Classes & Objects
- Instance methods

How do we find our data?

Finding a single record by its ID

```
id = 1
```

```
found_post = Post.find(id)
```

Finding a single record by another field

```
found_user = User.find_by({email: "email@email.com"})
```

How do we find our data?

Finding all records

```
posts = Post.all
```

Filtering the results

```
posts = Post.where({user_id: 1})
```

What is a Class?

A **class** is like a blueprint. It isn't the “thing” itself, but is a plan for how to build an actual, concrete instance of that thing.

An instance made from the **class** is called an **object**.



Creating an object

```
class House  
  # more to come inside  
end
```

```
house1 = House.new  
house2 = House.new
```

An **object** mixes data (variables) with functionality (methods).



Giving our class special powers

```
class House < ActiveRecord::Base  
  # more to come inside  
end
```

Take our class **House** and give it all the functionality found inside another class, **ActiveRecord::Base**.

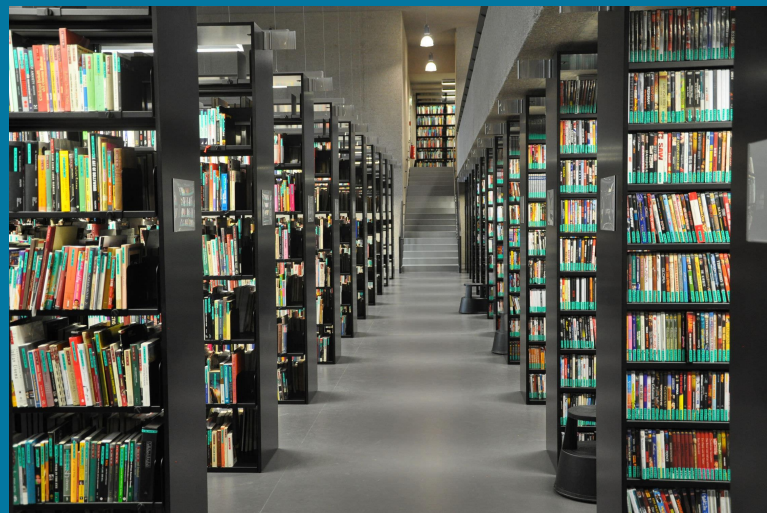
Now we can use **House** to interact with the **houses** table in our database.

ActiveRecord Relationships

Imagine a library full of books.

What is their relationship?

A library **has many** books.
A book **belongs to** a library.



ActiveRecord Relationships

Let's model these relationships in our code.

```
class Book < ActiveRecord::Base  
  belongs_to :library  
end
```

```
class Library < ActiveRecord::Base  
  has_many :books  
end
```

Using relationships to find data

```
tdot_library = Library.find(1)  
tdot_books = tdot_library.books.all
```

Broken down:

`tdot_library` = Instance of Library class (an object)
`books` = The has_many relationship of Library
`all` = A method to find all books (in this case).

The reverse is true

We can step through our relationships the other way too. Accessing a **belongs_to** relationship's object.

```
tdot_book = Book.find(1)  
tdot_book.library.name
```

This allows us to find the library that this book belongs to. After we access its name property.

Instance Methods

Let's give our objects a bit more functionality!

```
class Author < ActiveRecord::Base
  def initials
    "#{self.first_name[0]}#{self.last_name[0]}"
  end
end
```

Instance Methods

Time to use our new initials method.

```
robson = Author.new({first_name: "Dan", last_name:  
"Robson"})
```

```
robson.initials #=> "DR"
```

Your turn

Today we'll be finding data and showing that data in the view.