# In this lecture, we will discuss…

✧ Many-to-Many association

✧ Models and Migrations for a Many-to-Many

# Many-to-Many Association

✧ One person *can have many* hobbies

✧ One hobby can be *shared by many* people

✧ `habtm` (has_and_belongs_to_many)

✧ Need to create an extra (a.k.a. join) table (without a model, i.e. just a migration)

Convention: Plural model names separated by an underscore in <u>alphabetical</u> order

# Hobbies and Hobbies_People

```
~/advanced_ar$ rails g model hobby name
      invoke  active_record
      create    db/migrate/20150922154738_create_hobbies.rb
      create    app/models/hobby.rb
      invoke    test_unit
      create      test/models/hobby_test.rb
      create      test/fixtures/hobbies.yml
~/advanced_ar$ rails g migration create_hobbies_people person:references hobby:references
      invoke  active_record
      create    db/migrate/20150922154813_create_hobbies_people.rb
```

# Hobbies_People Migration

**FOLDERS**

- ▼ 📂 advanced_ar
  - ▶ 📁 app
  - ▶ 📁 bin
  - ▶ 📁 config
  - ▼ 📂 db
    - ▼ 📂 migrate
      - 📄 20150908214851_create_people.rb
      - 📄 20150908221446_add_login_pass_to_people
      - 📄 20150908232650_create_personal_infos.rb
      - 📄 20150922141356_create_jobs.rb
      - 📄 20150922154738_create_hobbies.rb
      - 📄 20150922154813_create_hobbies_people.rb

**20150922154813_create_hobbies_people.rb**

```ruby
class CreateHobbiesPeople < ActiveRecord::Migration
  def change
    create_table :hobbies_people, id: false do |t|
      t.references :person, index: true, foreign_key: true
      t.references :hobby, index: true, foreign_key: true
    end
  end
end
```

```
~/advanced_ar$ rake db:migrate
== 20150922154738 CreateHobbies: migrating ===========
-- create_table(:hobbies)
   -> 0.0009s
== 20150922154738 CreateHobbies: migrated (0.0010s) =======

== 20150922154813 CreateHobbiesPeople: migrating ========
-- create_table(:hobbies_people, {:id=>false})
   -> 0.0014s
== 20150922154813 CreateHobbiesPeople: migrated (0.0014s) =====
```

No need for primary key, since this is a join table

# Many-to-Many in the DB

```
~/advanced_ar$ rails db
SQLite version 3.8.5 2014-08-15 22:37:57
Enter ".help" for usage hints.
sqlite> .schema %hobbies%
CREATE TABLE "hobbies" ("id" INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, "name" varchar, "created_at" datetime NOT NULL, "updated_at" datetime NOT NULL);
CREATE TABLE "hobbies_people" ("person_id" integer, "hobby_id" integer);
CREATE INDEX "index_hobbies_people_on_person_id" ON "hobbies_people" ("person_id");
CREATE INDEX "index_hobbies_people_on_hobby_id" ON "hobbies_people" ("hobby_id");
```

# Person and Hobby Models

FOLDERS

- ▼ 📂 advanced_ar
  - ▼ 📂 app
    - ▶ 📁 assets
    - ▶ 📁 controllers
    - ▶ 📁 helpers
    - ▶ 📁 mailers
    - ▼ 📂 models
      - ▶ 📁 concerns
      - 📄 .keep
      - 📄 hobby.rb
      - 📄 job.rb
      - 📄 person.rb

**person.rb**

```ruby
class Person < ActiveRecord::Base
    has_one :personal_info, dependent: :destroy
    has_many :jobs
    has_many :my_jobs, class_name: "Job"
    has_and_belongs_to_many :hobbies
end
```

**hobby.rb**

```ruby
class Hobby < ActiveRecord::Base
    has_and_belongs_to_many :people
end
```

# Person and Hobby in Action

```
irb(main):002:0> josh = Person.find_by first_name: "Josh"
=> #<Person id: 32, first_name: "Josh", age: 57, last_name: "Oreck", created_at: "2015-09-22 15:06:15", updated_at: "2015-09-22 15:
s: "password1">
irb(main):003:0> lebron = Person.find_by first_name: "LeBron"
=> #<Person id: 35, first_name: "LeBron", age: 30, last_name: "James", created_at: "2015-09-22 15:06:15", updated_at: "2015-09-22 1
ass: "need more rings">
irb(main):004:0> programming = Hobby.create name: "Programming"
=> #<Hobby id: 3, name: "Programming", created_at: "2015-09-22 16:18:52", updated_at: "2015-09-22 16:18:52">
irb(main):005:0> josh.hobbies << programming; lebron.hobbies << programming
=> #<ActiveRecord::Associations::CollectionProxy [#<Hobby id: 3, name: "Programming", created_at: "2015-09-22 16:18:52", updated_at
irb(main):006:0> programming.people
=> #<ActiveRecord::Associations::CollectionProxy [#<Person id: 32, first_name: "Josh", age: 57, last_name: "Oreck", created_at: "20
d_at: "2015-09-22 15:06:15", login: "josh", pass: "password1">, #<Person id: 35, first_name: "LeBron", age: 30, last_name: "James",
5:06:15", updated_at: "2015-09-22 15:06:15", login: "bron", pass: "need more rings">]>
```

# Summary

✧ Many-to-Many contains 2 models and 3 migrations

✧ Join table needs to only exist in the DB, but not in Ruby code

## What's Next?

✧ Rich Many-to-Many association