

Rails Associations & Migrations

IIC2143 Ingeniería de Software - Ayudantía 5

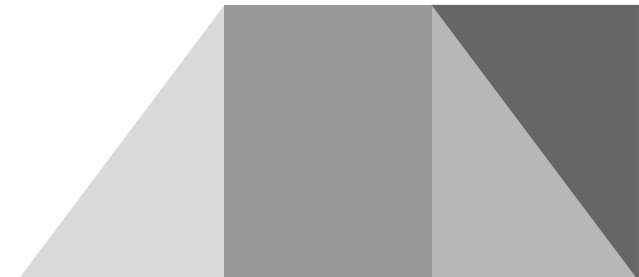
Associations

Associations

Is a connection between two Active Record models.

Why do we need associations between models?

Because they make common operations simpler and easier
in your code.





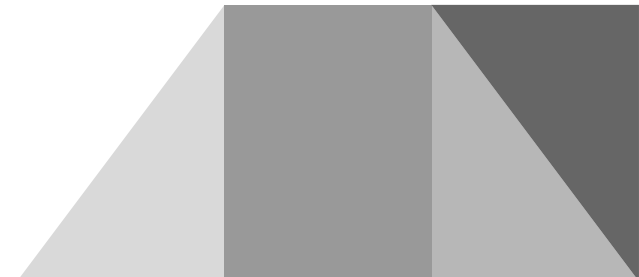
¿Qué tipos de asociaciones
conocen?

Tipos

2 The Types of Associations

Rails supports six types of associations:

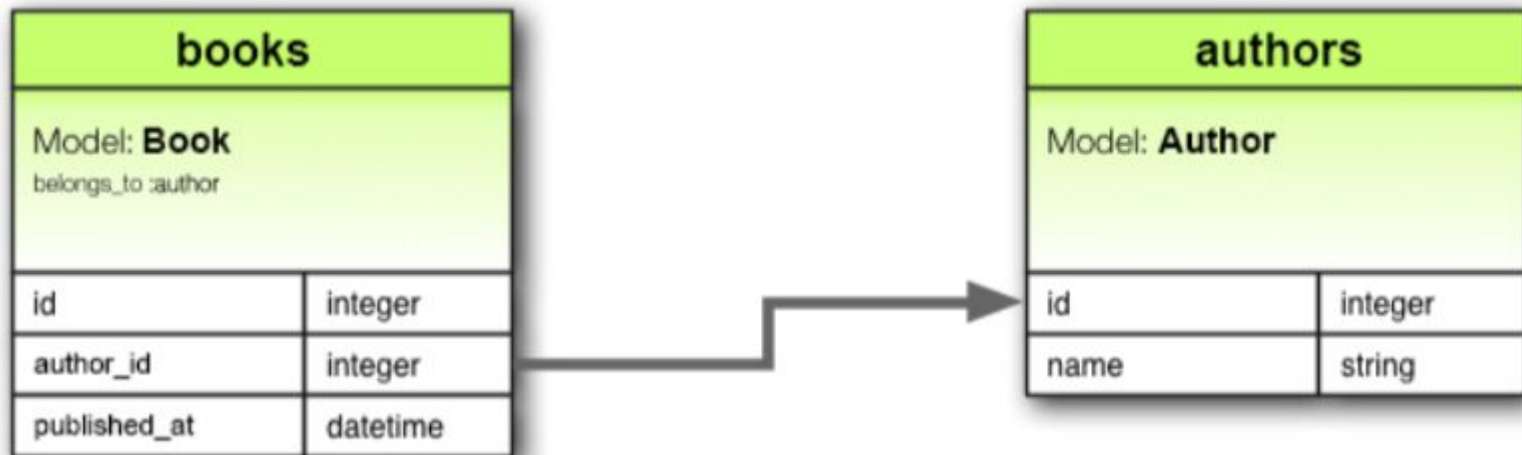
- `belongs_to`
- `has_one`
- `has_many`
- `has_many :through`
- `has_one :through`
- `has_and_belongs_to_many`





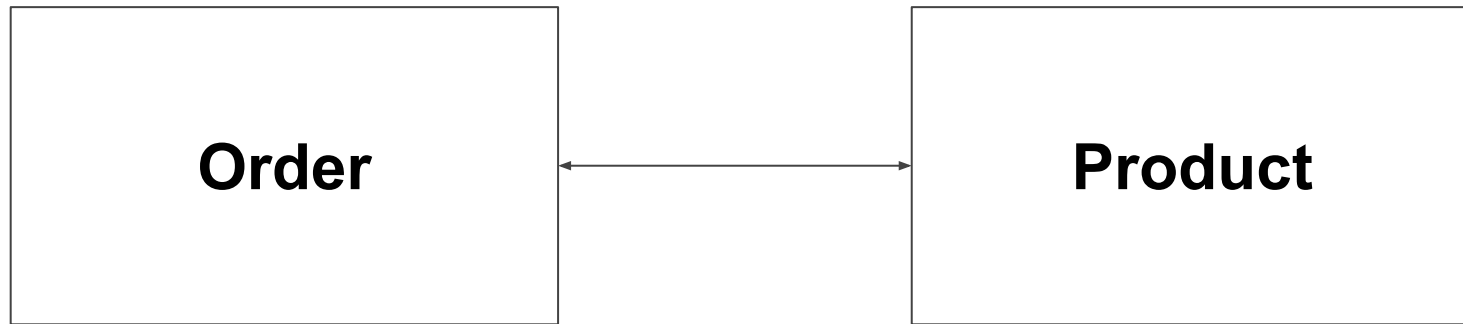
Algunos ejemplos

¿Cómo sería el código para el siguiente ejemplo?

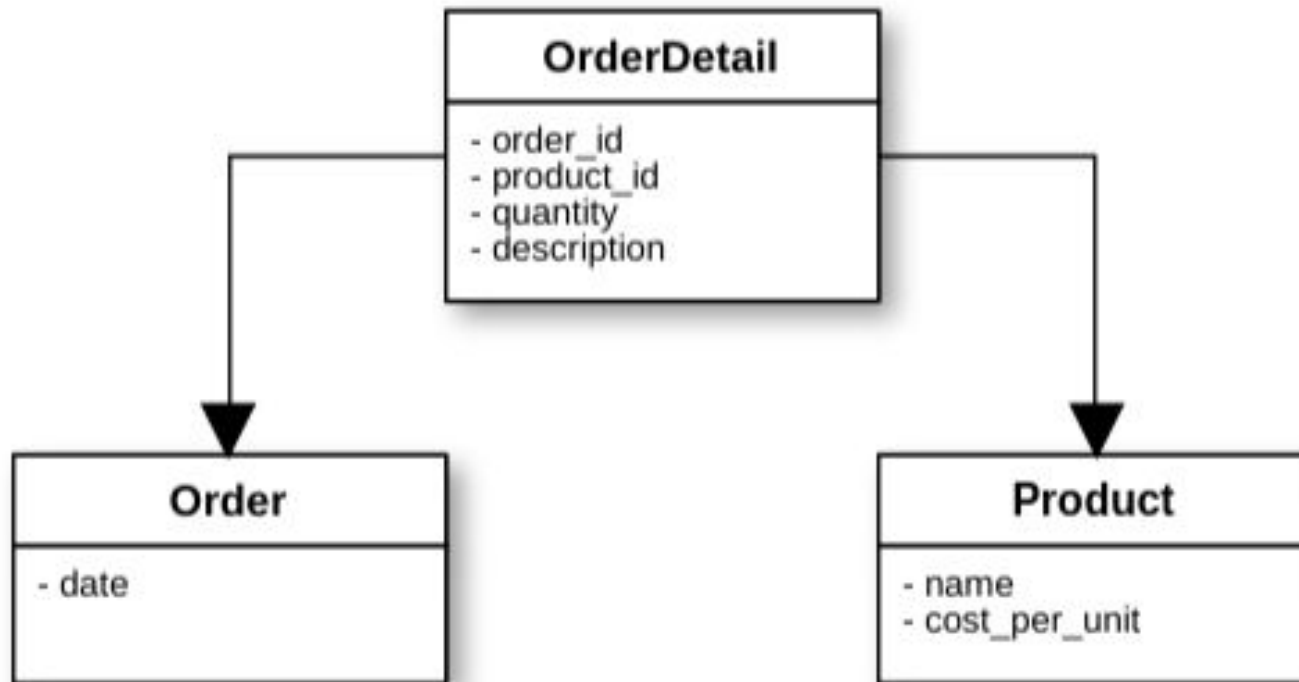


No recomendable: many to many

- Pongámonos en el siguiente caso, en que tenemos Ordenes y Productos



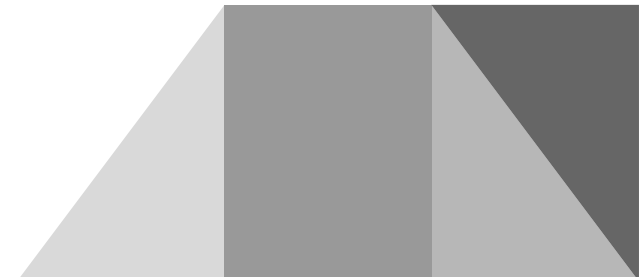
No recomendable: many to many



Migrations

¿Qué son las migraciones?

“Migrations are a convenient way to **alter your database schema over time** in a consistent and easy way”



¿Para qué nos sirven?

- Alterar nuestra base de datos
- Mantener distintos modelos en cada rama de trabajo
- Desligarnos del trabajo de la base de datos




Ejemplos

Docker-compose exec web...

```
$ rails generate migration AddPartNumberToProducts  
part_number:string:index
```

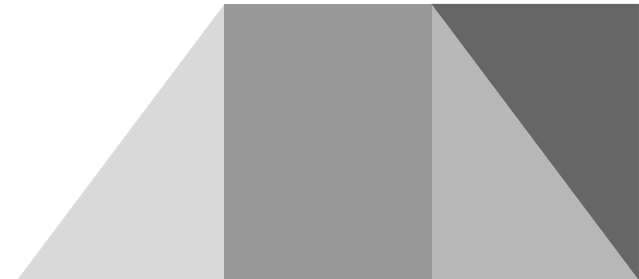
```
$ rails generate migration RemovePartNumberFromProducts  
part_number:string
```

```
$ rails generate migration AddUserRefToProducts user:references
```



Otros

- Cambiar el nombre de un atributo
- Agregar valores por defecto
- Cambiar valores por defecto
- Crear tablas
- etc...



Errores Comunes

Orden de Ejecución

```
• class CreateAlbums < ActiveRecord::Migration[5.1]
  def change
    create_table :albums do |t|
      t.string :name
      t.integer :year
      t.integer :total_duration
      t.references :artist, foreign_key: true
      t.timestamps
    end
  end
end
```

```
• class CreateArtists < ActiveRecord::Migration[5.1]
  def change
    create_table :artists do |t|
      t.string :name
      t.integer :birth_year
      t.integer :first_album_year
      t.timestamps
    end
  end
end
```


Editar Migraciones anteriores

```
• class CreateArtists < ActiveRecord::Migration[5.1]
  def change
    create_table :artists do |t|
      t.string :name
      t.integer :birth_year
      t.integer :first_album_year
      t.string :this_is_a_cool_new_attribute
      t.timestamps
    end
  end
end
```

Valores Default

Por defecto: **nil**

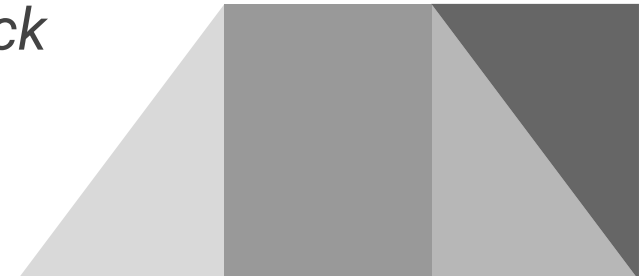
```
• class CreateAlbums < ActiveRecord::Migration[5.1]
  def change
    create_table :albums do |t|
      t.string :name
      t.integer :year
      t.integer :total_duration, default: 0
      t.references :artist, foreign_key: true
      t.timestamps
    end
  end
end
```

Tips

1. Hacer todos los cambios necesarios en la migración antes de migrar
2. Si agregamos atributos que son **obligatorios**... Agregar valor default en la migración
3. Si ya migré y me dí cuenta que debo agregar/borrar algo... Hacer una nueva migración para hacer ese cambio
4. Si queremos retroceder una migración, entonces podemos hacer:

\$ docker-compose exec web rake db:rollback

Deshace la última migración



Referencias

Associations:

- https://guides.rubyonrails.org/association_basics.html

Migrations:

- https://edgeguides.rubyonrails.org/active_record_migrations.html

