Understand Polymorphic in Ruby on Rails

A more advanced on associations is the polymorphic association. With polymorphic associations, a model cn belong to more than one other model, on a single association. For example, you might have like model that belongs to either an Book model or to a comment model.

First create model Book

```
$ rails generate model Book user:integer text:string
```

Create Model Comment

```
$ rails generate model Comment user:integer text:string
```

Create model Like

```
$ rails generate model Like
```

Check the migration of the model Like

```
class CreateLikes < ActiveRecord::Migration
    def change
        create_table :likes do |t|
        t.integer :user
        t.integer :likeable_id
        t.string :likeable_type

        t.timestamps
        end
    end
end
```

And here the model associations

```
class Like < ActiveRecord::Base
    belongs_to :likeable, :polymorphic => true
end

class Book < ActiveRecord::Base
    has_many :likes, :as => likeable
```

```
end

class Comment < ActiveRecord::Base
    has_many :likes, :as => likeable
end
```

Don't forget the migration

```
$ rails db:migrate
```

Now lets test in rails console

```
$ rails c

2.5.0 :001 > u = User.last
2.5.0 :002 > b = Book.last
2.5.0 :003 > l = Like.new(user: u)
=> #<Like id: nil, user_id: 19, likeable_id: nil, likeable_type: nil,
created_at: nil, updated_at: nil>
2.5.0 :004 > l.likeable = b
2.5.0 :005 > l
=> #<Like id: nil, user_id: 19, likeable_id: 19, likeable_type: "Book",
created_at: nil, updated_at: nil>
2.5.0 :006 > b.likes.count
=> # 1


2.5.0 :001 > u = User.last
2.5.0 :002 > c = Comment.last
2.5.0 :003 > l = Like.new(user: u)
=> #<Like id: nil, user_id: 19, likeable_id: nil, likeable_type: nil,
created_at: nil, updated_at: nil>
2.5.0 :004 > l.likeable = c
2.5.0 :005 > l
=> #<Like id: nil, user_id: 19, likeable_id: 30, likeable_type: "Comment",
created_at: nil, updated_at: nil>
2.5.0 :006 > c.likes.count
=> # 1
```

and thats it !!