



New Model:Visit

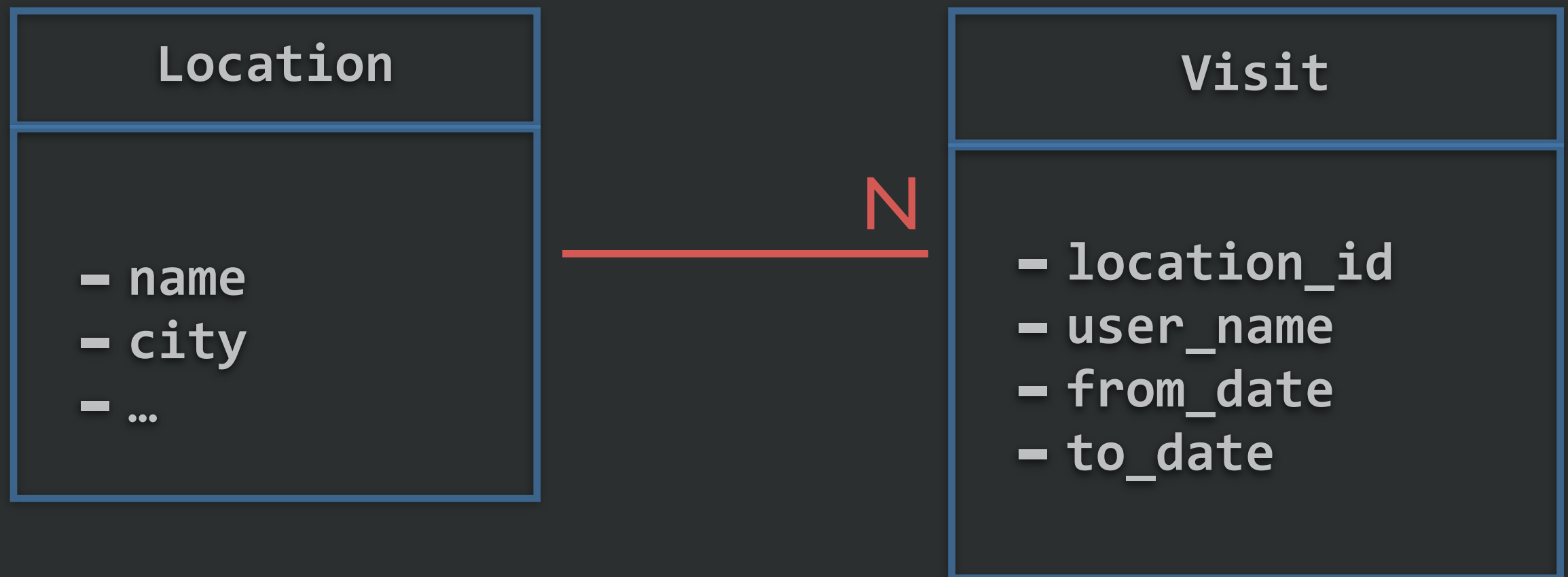
Sergio Revilla
@elrepicante

based on Martin Tschischauskas (@martynasc) slides

Model:Visit

Visit
<ul style="list-style-type: none">- location_id- user_name- from_date- to_date

Model:Visit



Model:Visit

```
$ rails g model visit
```

```
invoke active_record
create db/migrate/20140629134403_create_visits.rb
create app/models/visit.rb
invoke rspec
create spec/models/visit_spec.rb
```

Model:Visit

/models/visit.rb

```
class Visit < ActiveRecord::Base  
end
```

Exercise!

/db/migrate/xxxxxx_create_visits.rb

```
class CreateVisits < ActiveRecord::Migration
  def change
    create_table :visits do |t|

      t.timestamps
    end
  end
end
```

Model:Visit

/db/migrate/xxxxxx_create_visits.rb

```
class CreateVisits < ActiveRecord::Migration
  def change
    create_table :visits do |t|
      t.integer :location_id
      t.string :user_name
      t.datetime :from_date
      t.datetime :to_date
      t.timestamps
    end
  end
end
```

Model:Visit

```
$ rake db:migrate
```

```
== 20140629134403 CreateVisits: migrating =====  
-- create_table(:visits)  
   -> 0.0099s  
== 20140629134403 CreateVisits: migrated (0.0106s) =====
```

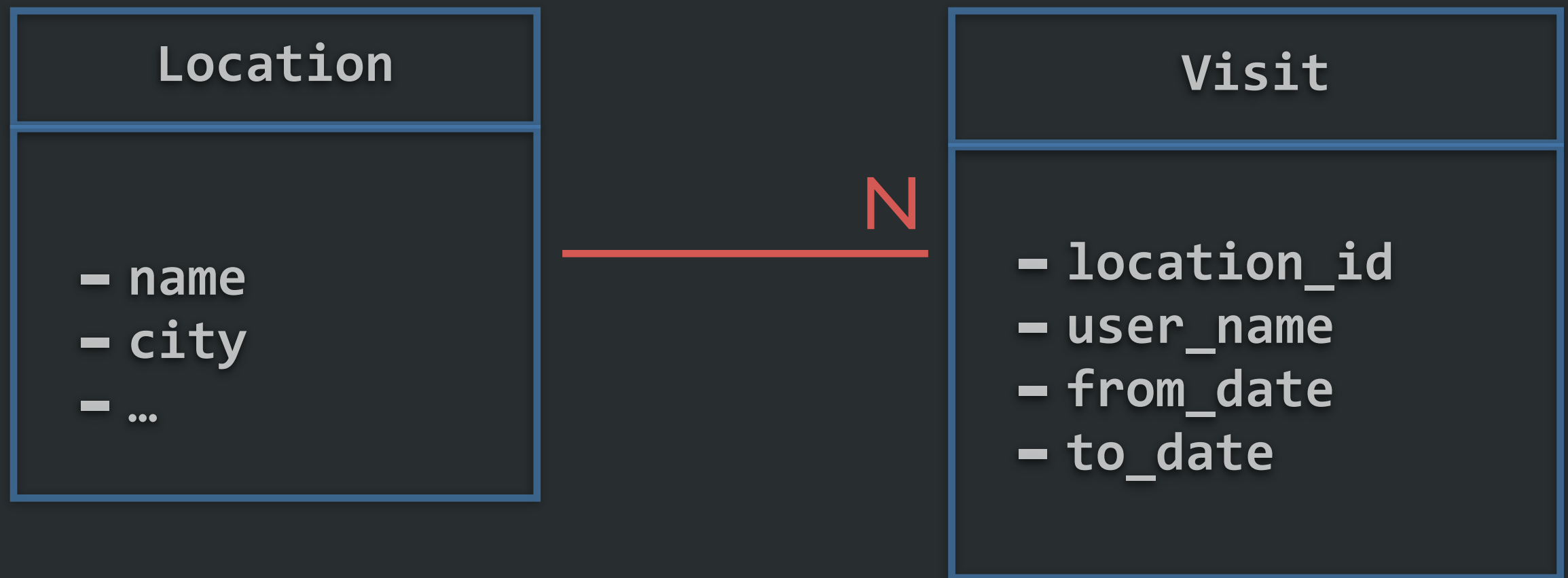

ActiveRecord Associations

ActiveRecord Associations

Provide an easy DSL to deal with relations between models

Allow to manage dependencies, such as cascade deletion

ActiveRecord Associations



ActiveRecord Associations

```
class Location < ActiveRecord::Base  
  has_many :visits  
end
```

```
class Visit < ActiveRecord::Base  
  belongs_to :location  
end
```

ActiveRecord Associations

When defined introduce lots of useful methods in our objects and classes

There are several types:

`has_many`

`belongs_to`

`has_one`

`has_and_belongs_to_many`

What does this all mean?

```
class Location < ActiveRecord::Base  
  has_many :visits  
end
```

```
class Visit < ActiveRecord::Base  
  belongs_to :location  
end
```

has_many

```
$ rails c
```

```
> location = Location.find 3
```

```
=> #<Location id: 3, name...
```

```
> location.visits
```

```
Visit Load (0.1ms) SELECT "visits".* FROM "visits" WHERE  
"visits"."location_id" = $1 ["location_id", 3]
```

```
> location.visits.count
```

```
=> 0
```

has_many

```
> visit = Visit.new  
> visit.user_name = "Martin"  
> visit.from_date = Time.now  
> visit.to_date = Time.now + 1.hour  
  
> visit.location = location  
> visit.save
```


has_many

```
INSERT INTO "visits" ("created_at", "from_date", "location_id",  
"to_date", "updated_at", "user_name") VALUES ($1, $2, $3, $4, $5,  
$6) RETURNING "id" [["created_at", "2014-06-29  
13:54:39.318370"], ["from_date", "2014-06-29 13:53:17.320384"],  
["location_id", 3], ["to_date", "2014-06-29 14:53:51.137556"],  
["updated_at", "2014-06-29 13:54:39.318370"], ["user_name",  
"Martin"]]
```

has_many

```
> location = Location.last  
> visit = location.visits.build  
=> #<Visit id: nil, location_id: 6, ...  
...  
> visit.save
```

has_many

```
INSERT INTO "visits" ("created_at", "from_date", "location_id",  
"to_date", "updated_at", "user_name") VALUES ($1, $2, $3, $4, $5,  
$6) RETURNING "id" [["created_at", "2014-06-29  
13:54:39.318370"], ["from_date", "2014-06-29 13:53:17.320384"],  
["location_id", 6], ["to_date", "2014-06-29 14:53:51.137556"],  
["updated_at", "2014-06-29 13:54:39.318370"], ["user_name",  
"Martin"]]
```

What is #visits?

`ActiveRecord::Associations::CollectionProxy`

Automatically applies a scope on the visits so that all the visits belong to the parent location

Very useful for creation and fetching

belongs_to

```
> visit = Visit.last
```

```
> visit.location
```

```
=> #<Location id: 6, name....
```

```
> visit.locations
```

```
NoMethodError: undefined method `locations' for #
```

belongs_to / has_many

- > `location.visits.where(user_name: "Martin").limit(2)`
- > `Visit.where(user_name: "Martin", location_id: 6).limit(2)`

Exercise #20

Update your `db/seeds.rb` with some Visits and associate them to a Location