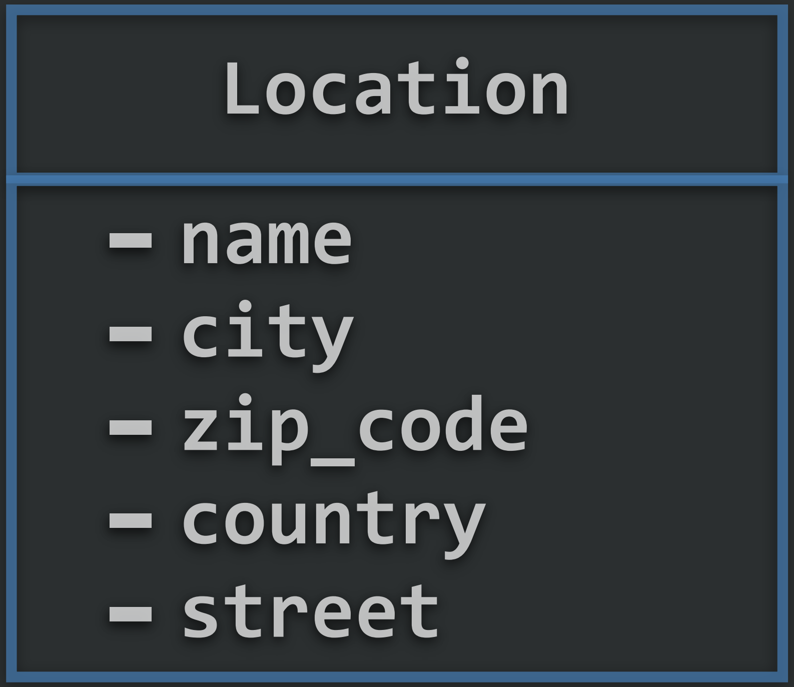
CRUD in ActiveRecord: Create & Read

Sergio Revilla
@elreplicante

based on Martin Tschischauskas (@martynasc) slides

Model: Location



Location

- name
- city
- zip_code
- country
- street

Create

Let’s use the rails console:

```
$ rails c
```

Create

```
> l = Location.new
> l.name = ‘Home’
> l.city = ‘Barcelona’
> l.save
```

Create

```
> l = Location.new name: ‘Home’,
                   city: ‘Barcelona’
> l.save
```

Create

```
> l = Location.create name: ’Home’,
                      city: ‘Barcelona’
```

Exercise #8

Create locations using the different ways.

Check the value of the attribute `id` and the timestamps.

Exercise #9

Check the value of the attribute `id` and the timestamps on a new instance:

```
> l = Location.new
> l.id
> l.created_at
> l.updated_at
```

Read

Retrieve a single object

Retrieve by primary key

#find

```
> l = Location.find(1)
```

When no elements found by id,
an exception is raised
`ActiveRecord::RecordNotFound`

Retrieve first element

#first

```
> l = Location.first
```

Retrieve last element

#last

```
> l = Location.last
```

Retrieve by attribute

#find_by

```
> l = Location.find_by name: ‘Home’
```

When no elements, methods return `nil`

#first!

#last!

#find_by!

When no elements, raise an exception
ActiveRecord::RecordNotFound

Retrieve a collection

Retrieve by conditions

#where

As a string

```
> Location.where “name=‘Home’”
```

Retrieve by conditions

#where

As a hash

```
> Location.where name: ‘Home’
```

Retrieve by conditions

#where

As a string with placeholders

```
> Location.where [“name=?”, ‘Home’]
```

Retrieve by conditions

#where

Concatenation

```
> Location.where(name: ’Home’).
   where(“created_at < ?”, Date.yesterday)
```

Retrieve by conditions

#where

Negation

```
> Location.where.not(“id < ?”, 4)
```

Retrieve with limits

#limit

```
> Location.limit(3).
    where(“city like ‘Bar%’”)
```

Retrieve with offset

#offset

```
> Location.limit(3).offset(2).
    where(“city like ‘Bar%’”)
```

Retrieve with order

#order

```
> Location.limit(10).
    order(“created_at DESC”)
```

Exercise #10

Write the following queries:

1. Get the last 10 locations created in the last month ordered by alphabetical name

2. Get the locations updated 1 minute ago

Solution #10

```
from = Time.now.midnight - 1.month
Location.order("name ASC").limit(10).
 where("created_at > ?”, from)
```

Solution #10

```
from = Time.now.midnight - 1.minute
to   = Time.now.midnight

Location.where(updated_at: from..to)
```

Exercise #11

Write a method #iron_find equivalent to #find, with the same parameters and the same behaviour

Solution #11

```ruby
class Location < ActiveRecord::Base
  def self.iron_find(id)
    where(id: id).first
  end
end
```