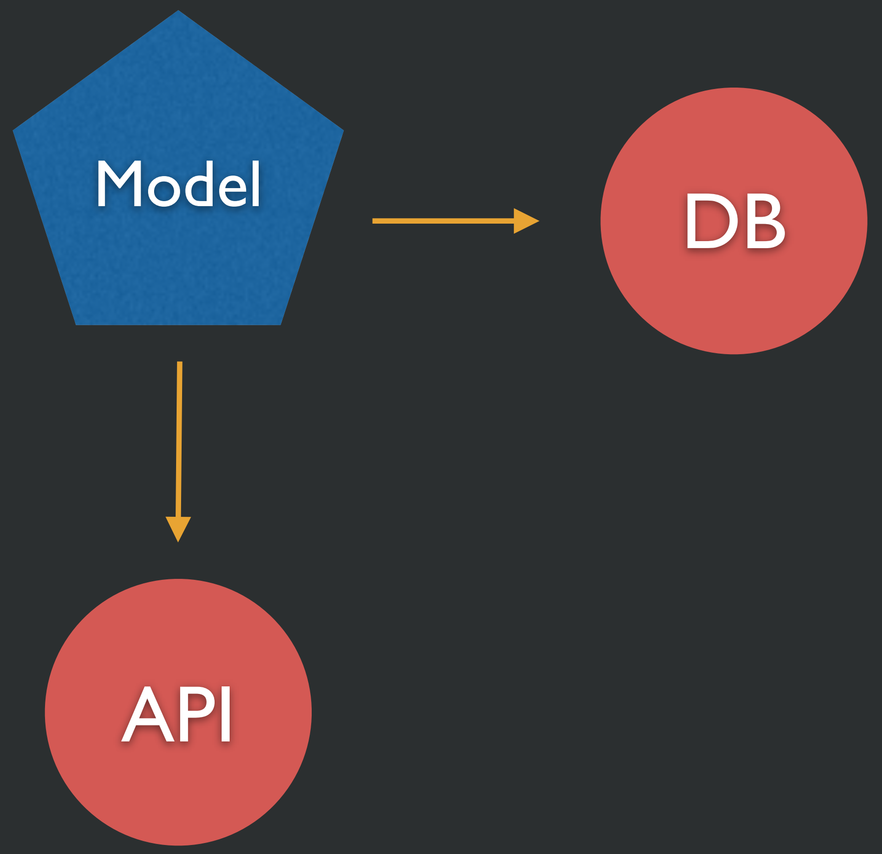
IRON
HACK

Model Validations

Sergio Revilla
@elreplicante

based on Martin Tschischauskas (@martynasc) slides

MVC: Models



1. Deal with data
2. Implement business logic rules

Deal with data

- Create, Read, Update, Delete (CRUD) from
  - Database
  - API
  - Filesystem
- Ensure data format is valid
    - Implement validations
      - i.e. length of a name is shorter than 50 chars

Business Logic Rules

Calculate the letter of the Spanish ID (NIF)

Calculate the VAT of a shopping cart

Generate reports

Single responsibility principle: an object should have only a single responsibility

Logic rules that ensure the data has proper **format** and **consistency**

Format validations

a valid email address

a web domain

a product price

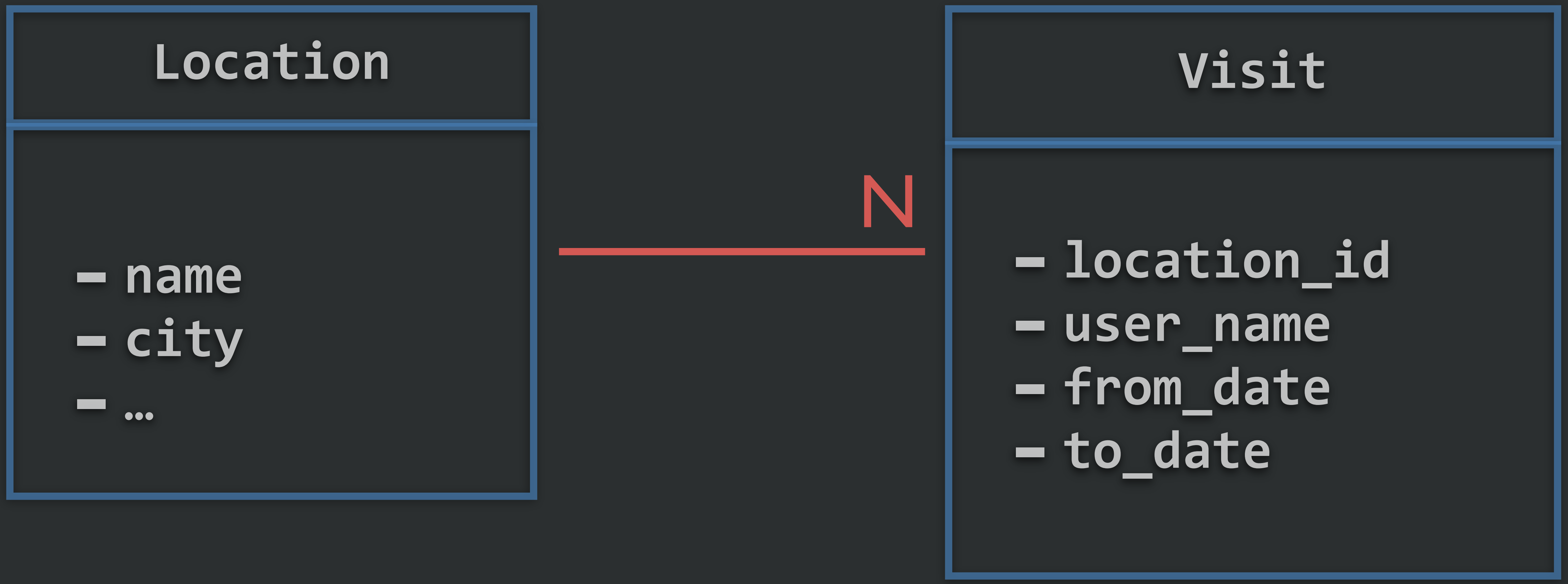
a time object

Consistency validations

a visit should belong to a location

a visit should occur in the future

Model: Visit



Location

- name
- city
- …

N

Visit

- location_id
- user_name
- from_date
- to_date

ActiveRecord Validations

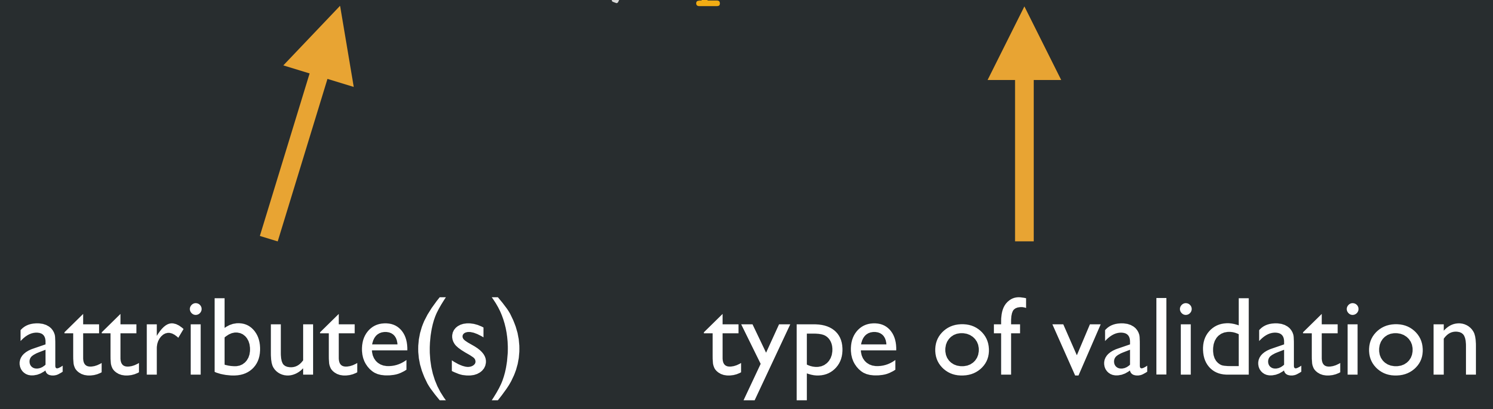
ActiveRecord validations

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

  validates :name, presence: true
end
```

```
> location = Location.new
=> #<Location id: nil, name: nil …
> location.valid?
=> false
> location.name = “Home”
> location.valid?
=> true
```

ActiveRecord validations

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

  validates :name, presence: true
end
```



attribute(s)

type of validation

ActiveRecord validations

```
validates :attribute, presence: true
```

Validates the presence of an attribute

ActiveRecord validations

```
validates :attribute, uniqueness: true
```

Validates the value of an attribute is unique

ActiveRecord validations

```
validates :attribute, length: {minimum: 3}
validates :attribute, length: {maximum: 10}
validates :attribute, length: {in: 1..100}
validates :attribute, length: {is: 42}
```

Validates the length of an attribute

IRON
HACK

ActiveRecord validations

```
validates :attribute, format: {with: /\A[0-9]/}
```

Validates the format of an attribute with a regular expression

ActiveRecord validations

```
  validates :attribute, inclusion: { in: [‘value 1’,
‘value 2’, ‘value 3’]}
```

Validates the value of an attribute to be in a closed list of values

ActiveRecord validations

```
  validates :attribute, exclusion: { in: [‘value 1’,
‘value 2’, ‘value 3’]}
```

Validates the value of an attribute to not be in a closed list of values

ActiveRecord validations

```
validates :attribute, numericality: true
```

Validates the value of an attribute to be a number

Exercise #25

Add the following validations to the model `Location`:

1. validate presence of `name`
2. validate presence of `city`
3. validate length of `name` to be a maximum of 30 characters

Exercise #25b

- Fix existing seeds

Exercise #26

Add the following validations to the model `Visit`:

1. from_date and to_date presence
2. validate presence of `user_name`
3. validate it's associated to a `location`

Exercise #26b

- Fix existing seeds