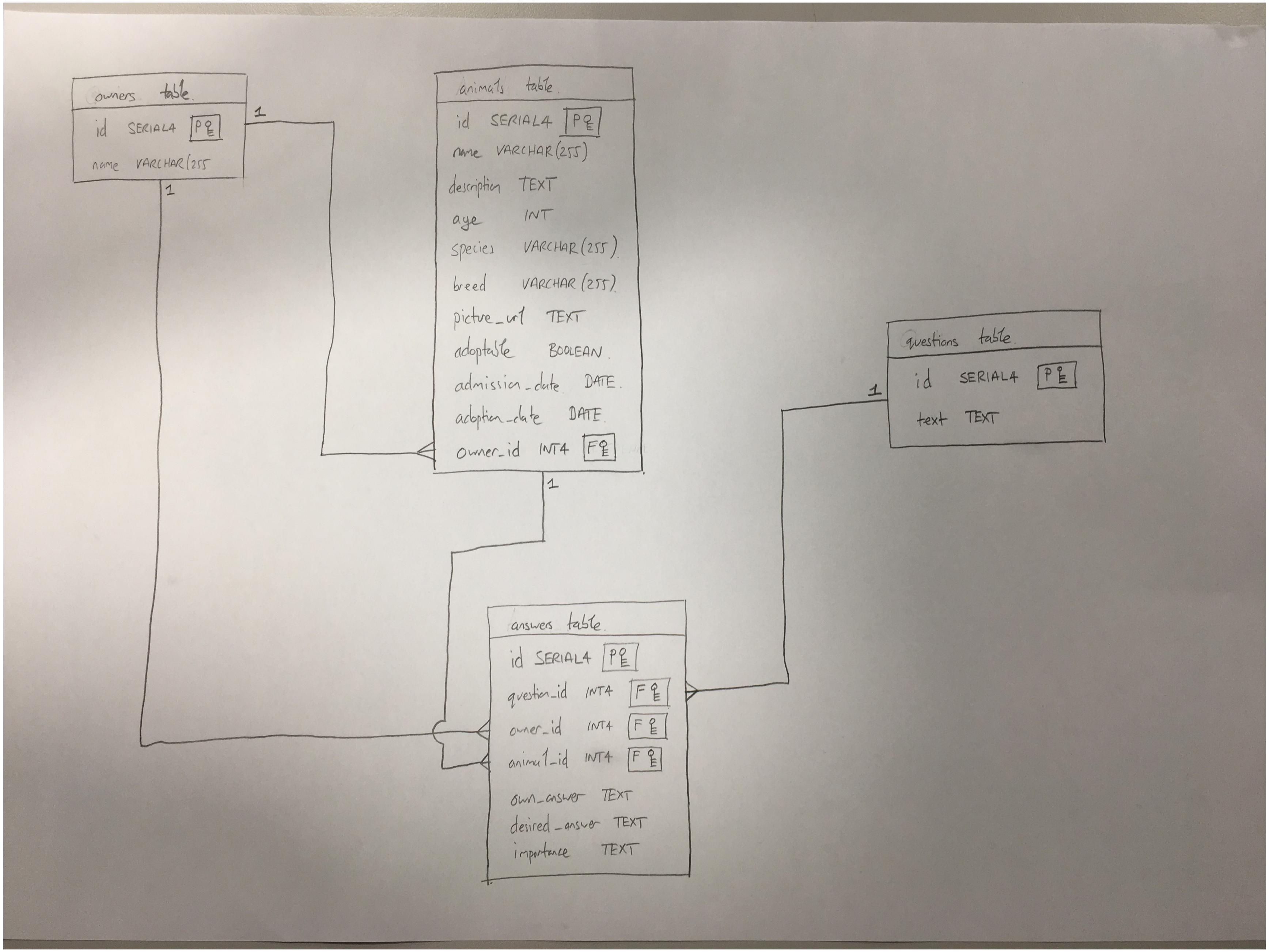
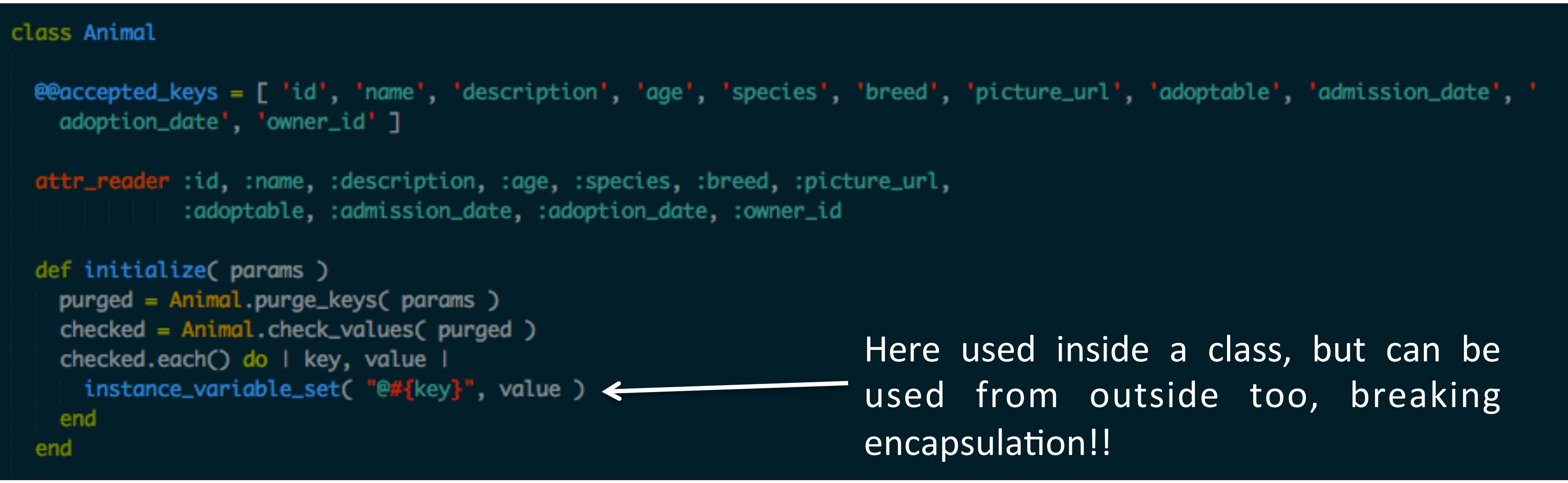
Lonely Hearts Animal Shelter

Database diagram



Demonstration

Nice (but potentially abusable code)



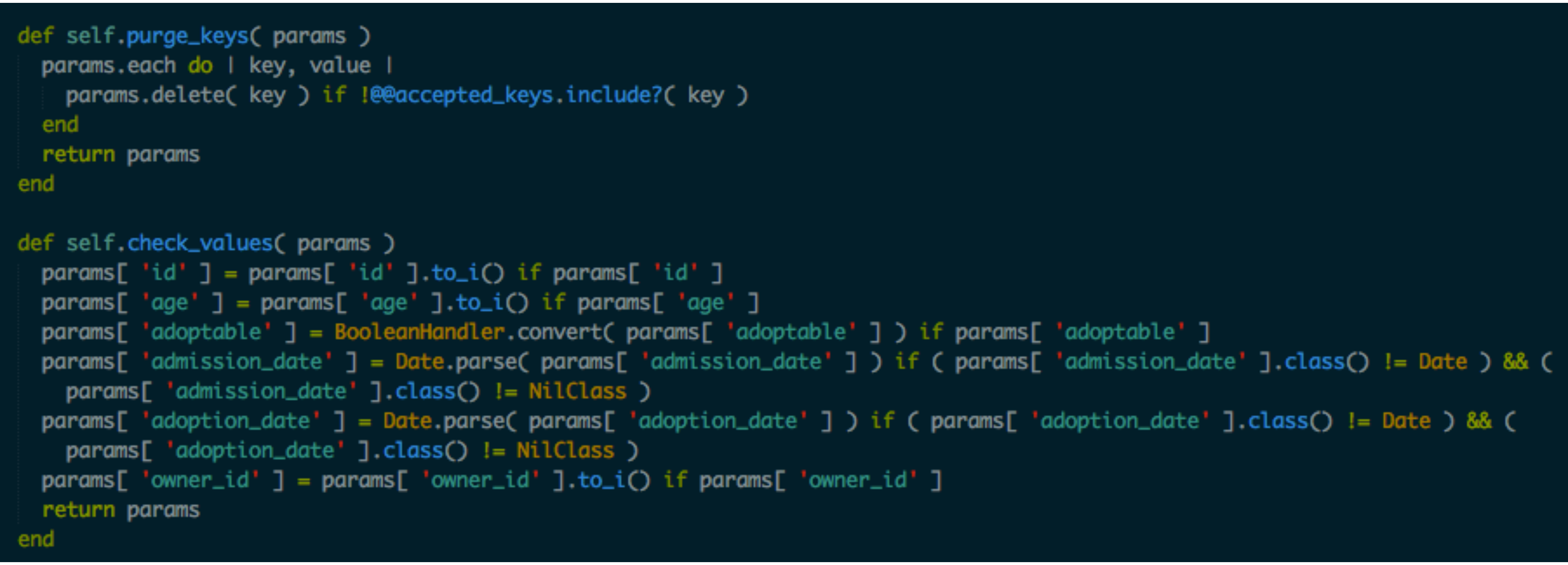
```ruby
class Animal

  @@accepted_keys = [ 'id', 'name', 'description', 'age', 'species', 'breed', 'picture_url', 'adoptable', 'admission_date', '
    adoption_date', 'owner_id' ]

  attr_reader :id, :name, :description, :age, :species, :breed, :picture_url,
              :adoptable, :admission_date, :adoption_date, :owner_id

  def initialize( params )
    purged = Animal.purge_keys( params )
    checked = Animal.check_values( purged )
    checked.each() do | key, value |
      instance_variable_set( "@#{key}", value )
    end
  end
```

Here used inside a class, but can be used from outside too, breaking encapsulation!!



```ruby
def self.purge_keys( params )
  params.each do | key, value |
    params.delete( key ) if !@@accepted_keys.include?( key )
  end
  return params
end

def self.check_values( params )
  params[ 'id' ] = params[ 'id' ].to_i() if params[ 'id' ]
  params[ 'age' ] = params[ 'age' ].to_i() if params[ 'age' ]
  params[ 'adoptable' ] = BooleanHandler.convert( params[ 'adoptable' ] ) if params[ 'adoptable' ]
  params[ 'admission_date' ] = Date.parse( params[ 'admission_date' ] ) if ( params[ 'admission_date' ].class() != Date ) && (
    params[ 'admission_date' ].class() != NilClass )
  params[ 'adoption_date' ] = Date.parse( params[ 'adoption_date' ] ) if ( params[ 'adoption_date' ].class() != Date ) && (
    params[ 'adoption_date' ].class() != NilClass )
  params[ 'owner_id' ] = params[ 'owner_id' ].to_i() if params[ 'owner_id' ]
  return params
end
```

Learning points

I was very reluctant to expand my database initially to include '*answers*' and '*questions*' tables, and instead set about essentially re-inventing the wheel in Ruby. Making use of pre-existing tools was sensible, and led to cleaner code (thanks Alex).

Later in the project I found a tension between user experience (and useability) vs. RESTful routing. For example, it would make RESTful sense to have an '*/animals/:id/answers/new*' route that returns a form to answer a **single** question for an animal. But the user would likely wish to answer **more** than one question per form page.

I'm uneasy about the existence of *instance_variable_get()* and *instance_variable_set()*…