In this lecture, we will discuss…

- Active Record, the ORM
- What conventions does Active Record assume?

Models: ORM

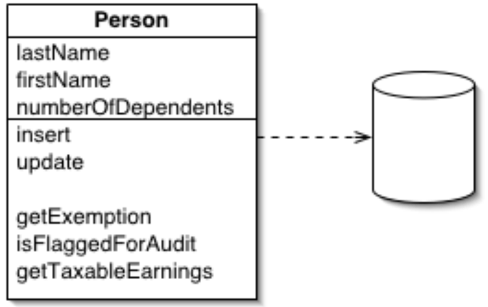
- ORM (Object-Relational Mapping)
  - Bridges the gap between relational databases, which are designed around mathematical Set Theory and Object-Oriented programming languages that deal with objects and their behavior.
  - Greatly simplifies writing code for accessing the database
- In Rails, the Model (usually) uses some ORM framework

Design pattern

Active Record

An object that wraps a row in a database table or view, encapsulates the database access, and adds domain logic on that data.

For a full description see P of EAA page **160**

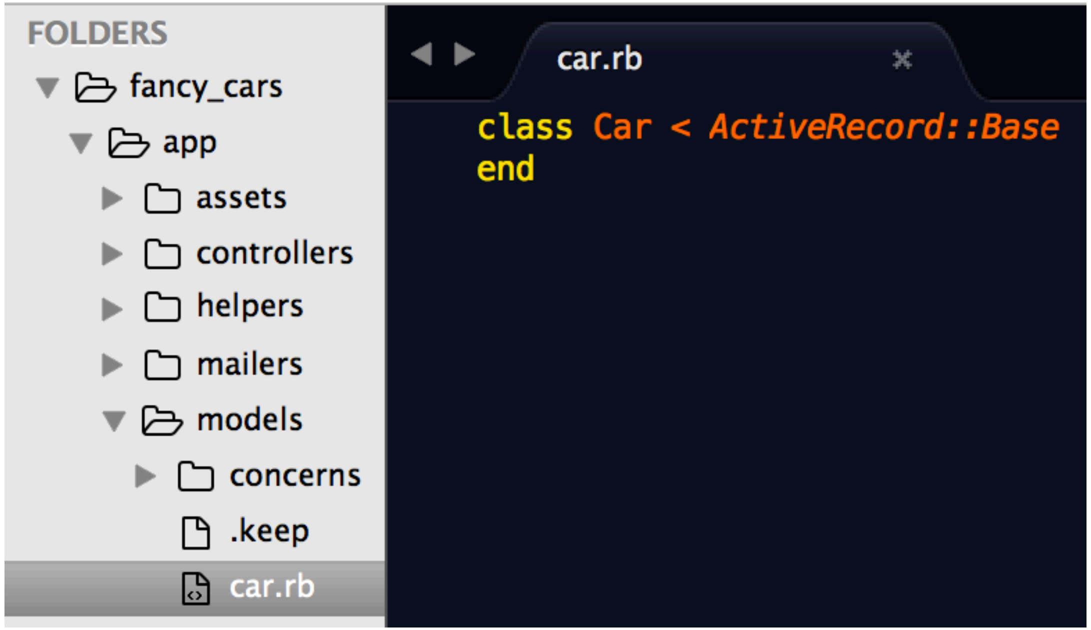


Martin Fowler

An object carries both data and behavior. Much of this data is persistent and needs to be stored in a database. Active Record uses the most obvious approach, putting data access logic in the domain object. This way all people know how to read and write their data to and from the database.

Active Record

- ***ActiveRecord*** is also the name of Rails' default ORM



```ruby
class Car < ActiveRecord::Base
end
```

Where is all the code?

Metaprogramming + Conventions

ActiveRecord

- Three Prerequisites:

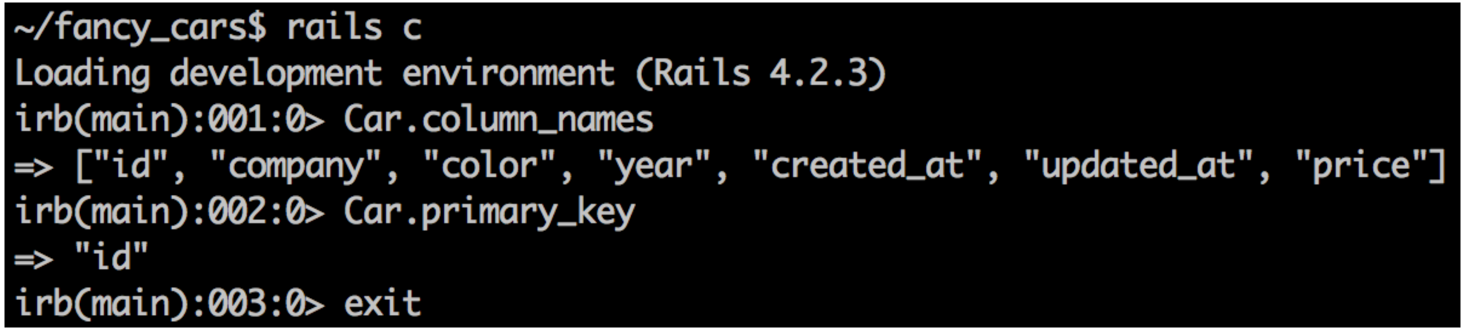
1. ActiveRecord has to know how to find your database (when Rails is loaded, this info is read from `config/database.yml` file)

2. (*Convention*) There is a table with a plural name that corresponds to `ActiveRecord::Base` subclass with a singular name

3. (*Convention*) Expects the table to have a primary key named `id`

Rails Console: rails c

- "IRB on steroids" with your Rails App loaded

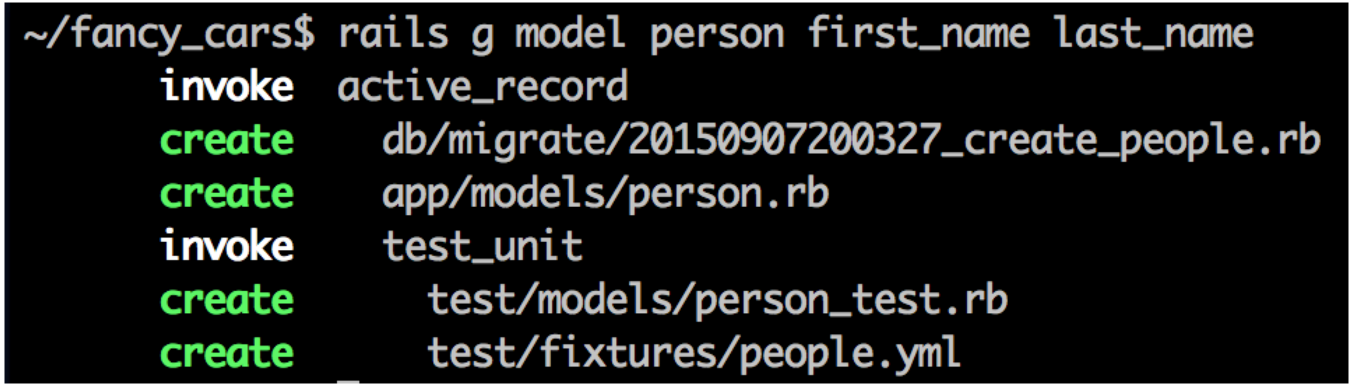


```
~/fancy_cars$ rails c
Loading development environment (Rails 4.2.3)
irb(main):001:0> Car.column_names
=> ["id", "company", "color", "year", "created_at", "updated_at", "price"]
irb(main):002:0> Car.primary_key
=> "id"
irb(main):003:0> exit
```

Class methods deal with the table as a whole, while instance methods deal with a particular row of the table…

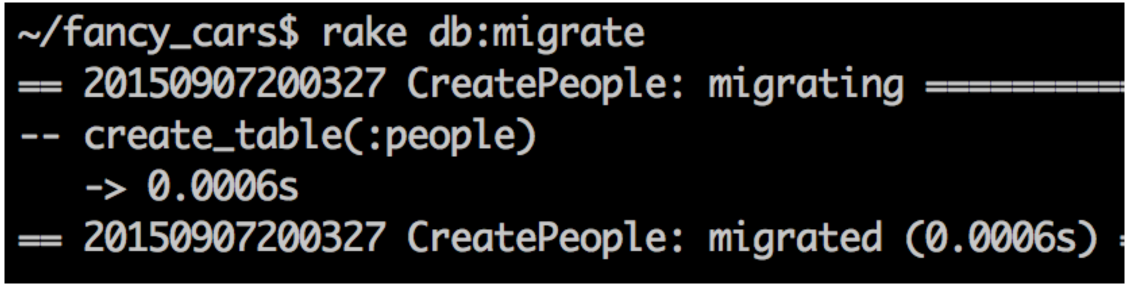
Model and Migration

- We saw the scaffold generator and the migration generator, but it turns out model has its own generator as well, which could also generate a migration



```
~/fancy_cars$ rails g model person first_name last_name
      invoke  active_record
      create    db/migrate/20150907200327_create_people.rb
      create    app/models/person.rb
      invoke    test_unit
      create      test/models/person_test.rb
      create      test/fixtures/people.yml
```

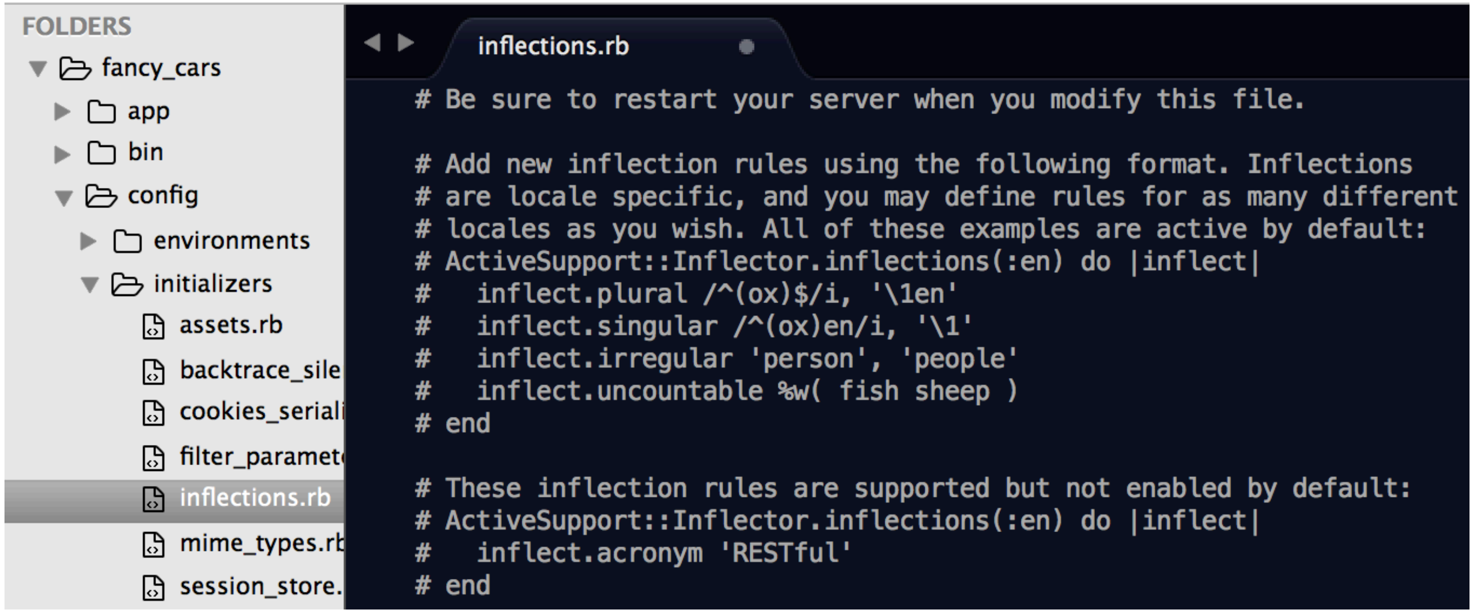
Model and Migration



```
~/fancy_cars$ rake db:migrate
== 20150907200327 CreatePeople: migrating ==========
-- create_table(:people)
   -> 0.0006s
== 20150907200327 CreatePeople: migrated (0.0006s)
```

Smart enough to know that the table name is **people** not persons

People Not Persons



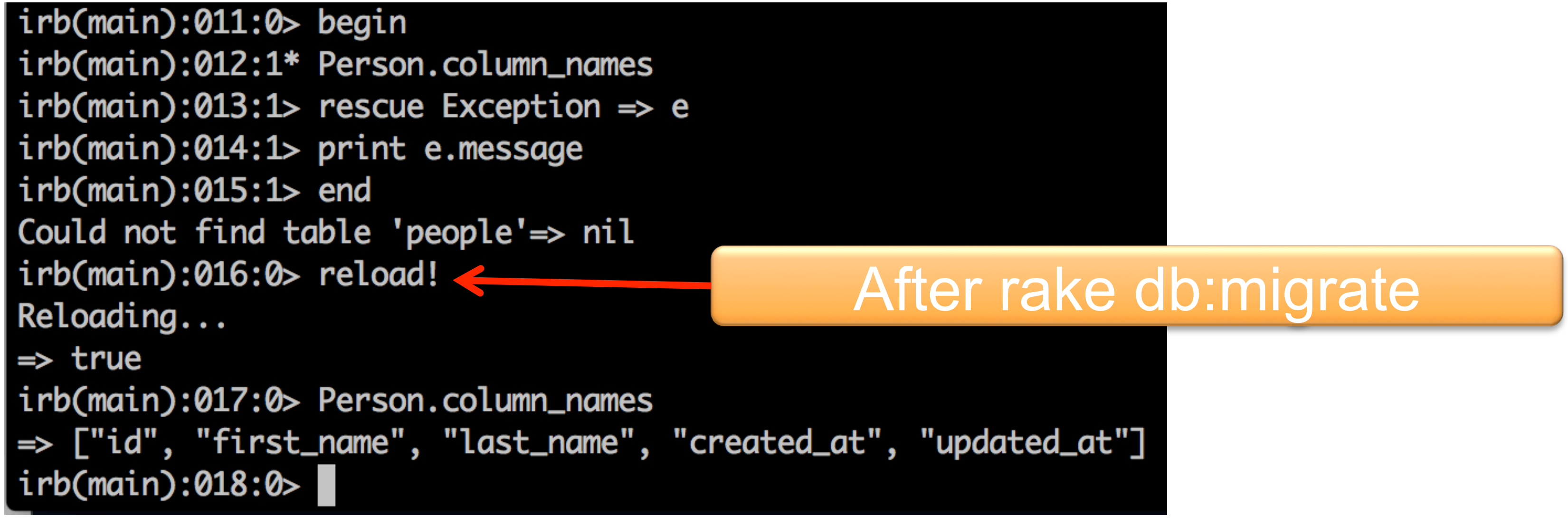
```
# Be sure to restart your server when you modify this file.

# Add new inflection rules using the following format. Inflections
# are locale specific, and you may define rules for as many different
# locales as you wish. All of these examples are active by default:
# ActiveSupport::Inflector.inflections(:en) do |inflect|
#   inflect.plural /^(ox)$/i, '\1en'
#   inflect.singular /^(ox)en/i, '\1'
#   inflect.irregular 'person', 'people'
#   inflect.uncountable %w( fish sheep )
# end

# These inflection rules are supported but not enabled by default:
# ActiveSupport::Inflector.inflections(:en) do |inflect|
#   inflect.acronym 'RESTful'
# end
```

Reloading Rails Console

- Don't have to kill rails console after a new migration – just call `reload`!



```
irb(main):011:0> begin
irb(main):012:1* Person.column_names
irb(main):013:1> rescue Exception => e
irb(main):014:1> print e.message
irb(main):015:1> end
Could not find table 'people'=> nil
irb(main):016:0> reload!
Reloading...
=> true
irb(main):017:0> Person.column_names
=> ["id", "first_name", "last_name", "created_at", "updated_at"]
irb(main):018:0>
```

Summary

✧ Active Record conventions:
- Class name is singular
- DB table name is plural
- Need to have an `id` primary key

What's Next?

✧ Active Record CRUD