Introduction to Ruby on Rails

Mostafa Ragab Arafa

Why Ruby on Rails?

- MVC – Model View Controller architecture

- Convention over Configuration – spend less time configuring things!

- DRY – Don't Repeat Yourself

- TDD – Test Driven Development

- jQuery – Javascript library

- SASS – Better CSS (variables &∓ conditionals!)

- CoffeeScript – Nicer/easier to read/write Javascript

- REST – REpresentational State Transfer – architecture for (stateless) networked applications

Model .. Active Record

- Object Relational Mapping (ORM)

- Convention over Configuration in Active Record

  - Naming Conventions

  - Schema Conventions

Model / Class	Table / Schema
Article	articles
LineItem	line_items
Deer	deers
Mouse	mice
Person	people

Model .. Active Record

- Creating Active Record Models

- CRUD: Reading and Writing Data

- Validations

  - validates :name, presence: true

- Callbacks

- Overriding the Naming Conventions

Migrations

- Domain-specific language for managing a database schema.

- Stored in files which are executed against any database that Active Record supports using *rake*.

- *rake db:migrate*

- *rake db:rollback*

Controller .. Action Controller

- Controller Naming Convention

- Methods and Actions

- Parameters

- Strong Parameters

- The flash

- Rendering XML and JSON data

- Filters

Controller .. Action Controller

- The request Object

Property of request	Purpose
host	The hostname used for this request.
domain(n=2)	The hostname's first n segments, starting from the right (the TLD).
format	The content type requested by the client.
method	The HTTP method used for the request.
get?, post?, patch?, put?, delete?, head?	Returns true if the HTTP method is GET/POST/PATCH/PUT/DELETE/HEAD.
headers	Returns a hash containing the headers associated with the request.
port	The port number (integer) used for the request.
protocol	Returns a string containing the protocol used plus "://", for example "http://".
query_string	The query string part of the URL, i.e., everything after "?".
remote_ip	The IP address of the client.
url	The entire URL used for the request.

Controller .. Action Controller

- The response Object

Property of response	Purpose
body	This is the string of data being sent back to the client. This is most often HTML.
status	The HTTP status code for the response, like 200 for a successful request or 404 for file not found.
location	The URL the client is being redirected to, if any.
content_type	The content type of the response.
charset	The character set being used for the response. Default is "utf-8".
headers	Headers used for the response.

Layouts and Rendering

- Layouts and views
- *render* method
- *redirect_to* method
- Structuring Layouts
  - Asset tags
  - *yield* and *content_for*
  - Partials
- Helpers

Routes

- The Rails router recognizes URLs and dispatches them to a controller's action. It can also generate paths and URLs, avoiding the need to hardcode strings in your views.

- Output for *resources :photos*

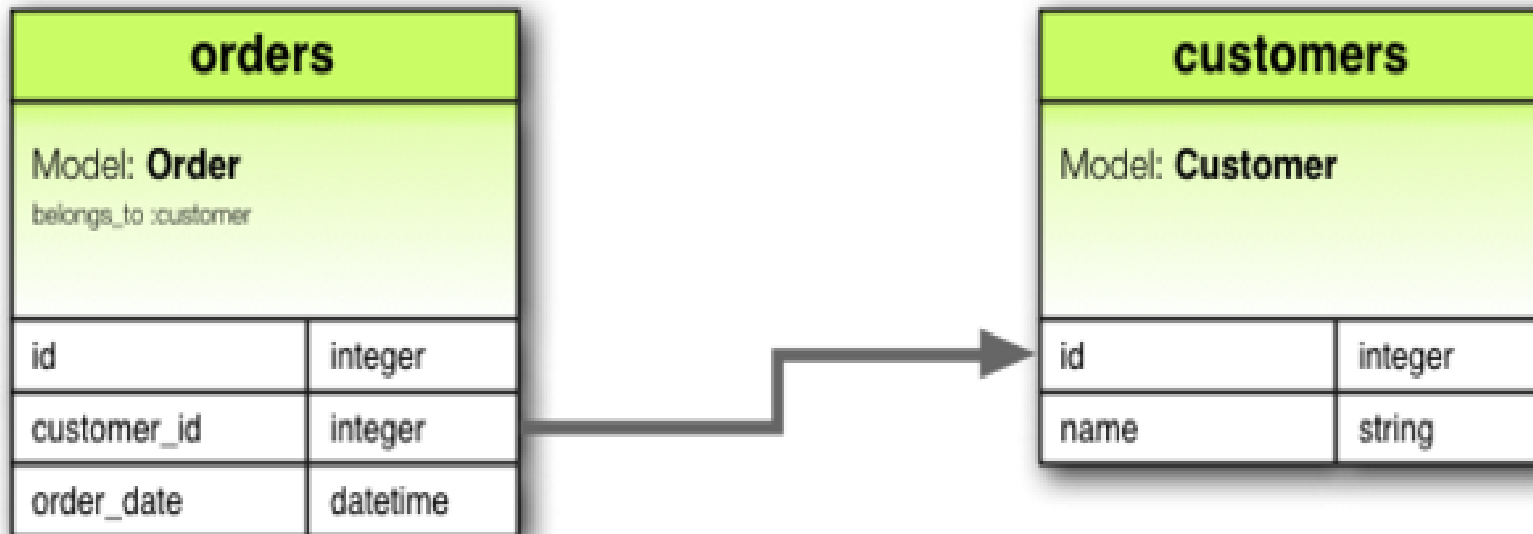| HTTP Verb	Path	Controller#Action	Used for
GET	/photos	photos#index	display a list of all photos
GET	/photos/new	photos#new	return an HTML form for creating a new photo
POST	/photos	photos#create	create a new photo
GET	/photos/:id	photos#show	display a specific photo
GET	/photos/:id/edit	photos#edit	return an HTML form for editing a photo
PATCH/PUT	/photos/:id	photos#update	update a specific photo
DELETE	/photos/:id	photos#destroy	delete a specific photo

Routes

- Path and URL Helpers

- Nested Resources

- *link_to* helper

- Custom RESTful actions (member vs collection)

- Creating Paths and URLs From Objects

- *rake routes*

Active Record Associations

- belongs_to
- has_one
- has_many
- has_many :through
- has_one :through
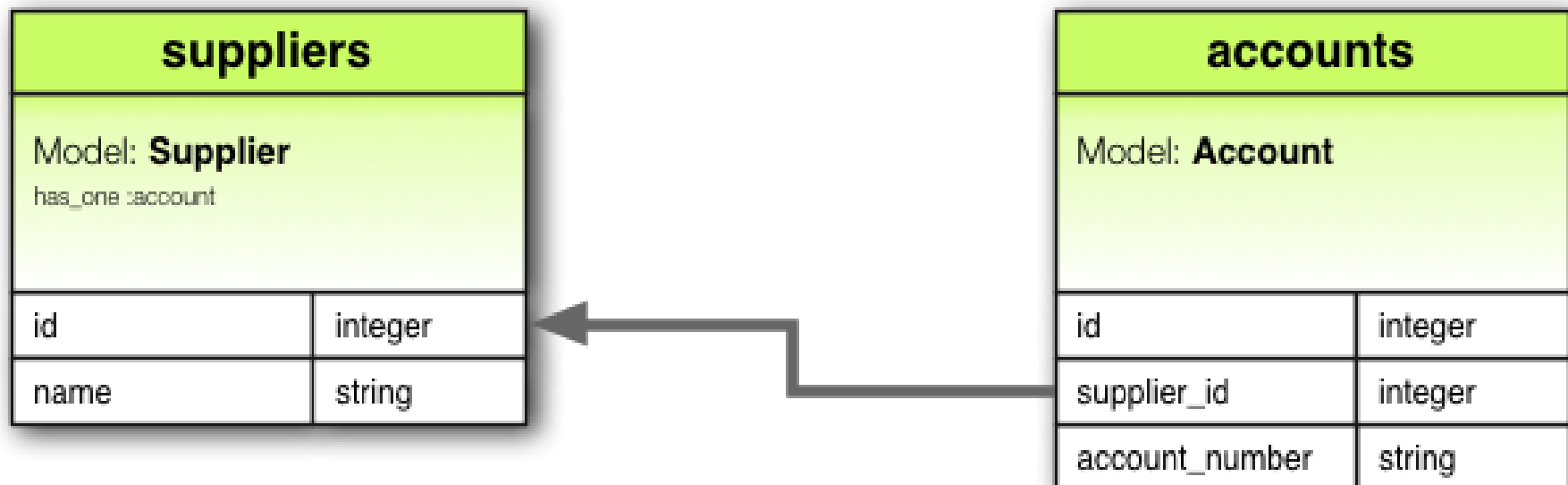- has_and_belongs_to_many

Active Record Associations

- *belongs_to* association


```
class Order < ActiveRecord::Base
    belongs_to :customer
end
```
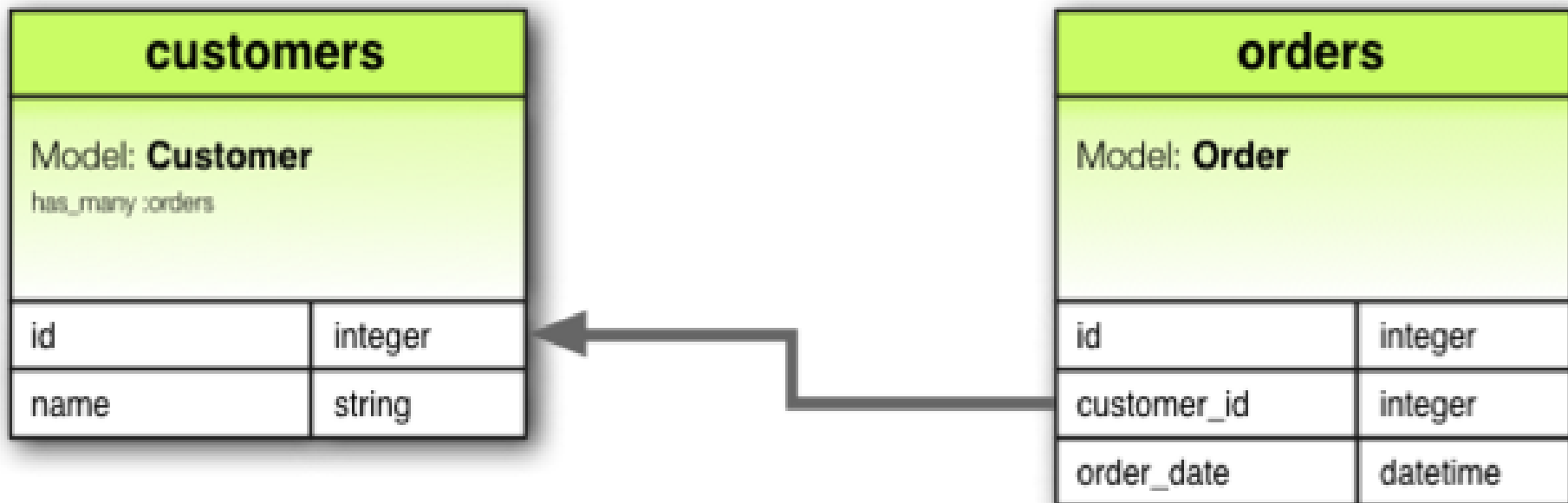
Active Record Associations

- *has_one* association


```
class Supplier < ActiveRecord::Base
  has_one :account
end
```

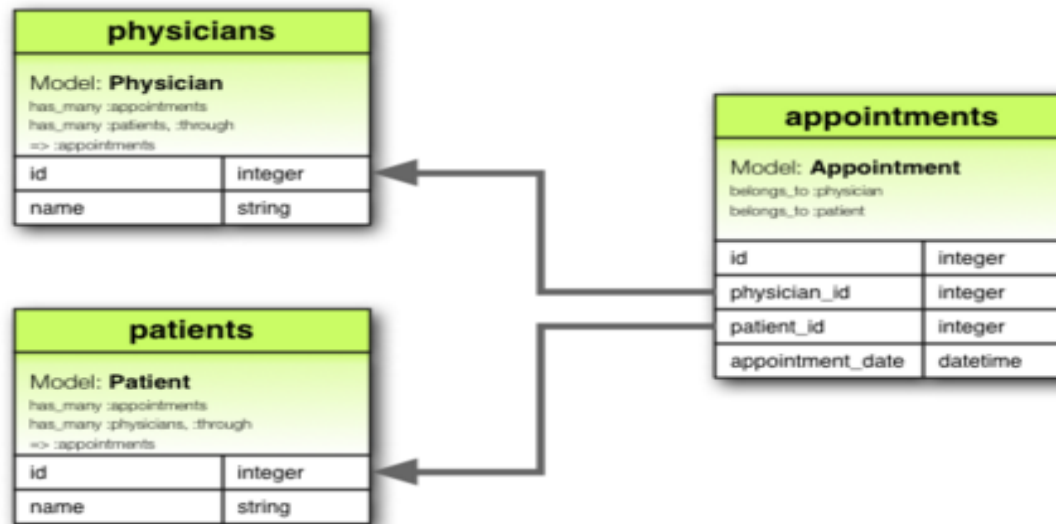# Active Record Associations

- *has_many* association


```
class Customer < ActiveRecord::Base
  has_many :orders
end
```

Active Record Associations

- *has_many :through* association


```
class Physician < ActiveRecord::Base
  has_many :appointments
  has_many :patients, :through => :appointments
end

class Appointment < ActiveRecord::Base
  belongs_to :physician
  belongs_to :patient
end

class Patient < ActiveRecord::Base
  has_many :appointments
  has_many :physicians, :through => :appointments
end
```
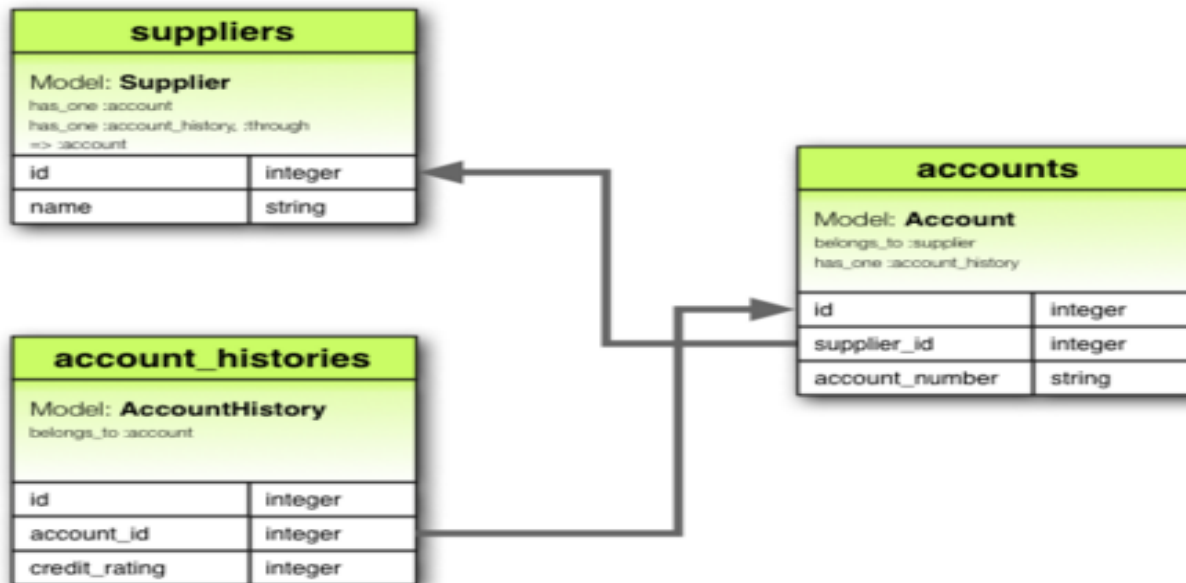
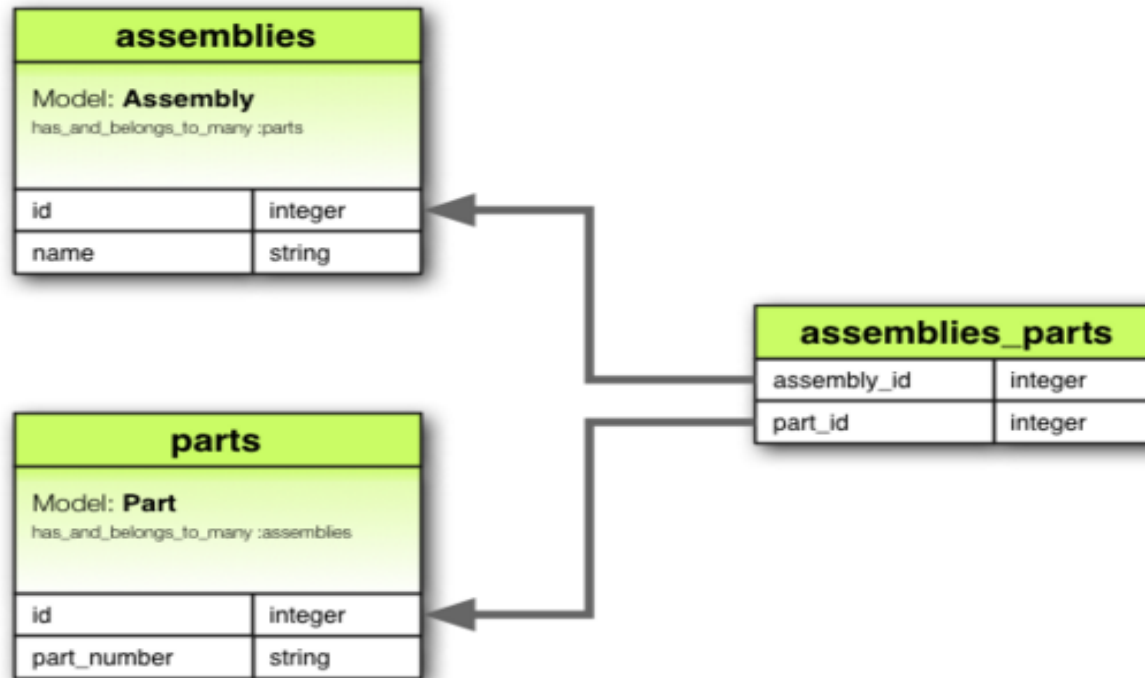# Active Record Associations

- *has_one :through* association


```
class Supplier < ActiveRecord::Base
   has_one :account
   has_one :account_history, :through => :account
end

class Account < ActiveRecord::Base
   belongs_to :supplier
   has_one :account_history
end

class AccountHistory < ActiveRecord::Base
   belongs_to :account
end
```

Active Record Associations

- *has_and_belongs_to_many* association


```
class Assembly < ActiveRecord::Base
  has_and_belongs_to_many :parts
end

class Part < ActiveRecord::Base
  has_and_belongs_to_many :assemblies
end
```

scaffold

- A full set generator of model, database migration for that model, controller to manipulate it, views to view and manipulate the data, and a test suite for each of the above.

- *rails generate scaffold HighScore game:string*

Bundler

- Bundler is the default gem dependency manager for Ruby projects

- In any rails project, all used gems must be listed in *Gemfile inside your project*

Rails Command Lines

- rails console

- rails server

- rake

- rails generate

- rails destroy

- rails dbconsole

- rails new app_name

Resources

- http://guides.rubyonrails.org/active_record_basics.html
- http://guides.rubyonrails.org/action_controller_overview.html
- http://guides.rubyonrails.org/layouts_and_rendering.html
- http://guides.rubyonrails.org/routing.html
- http://guides.rubyonrails.org/association_basics.html
- http://guides.rubyonrails.org/command_line.html

Thank You

Mostafa Ragab Arafa