

Лекция 12. Хранение данных. ORM. Модели Rails

- Принципы ORM
- Модели Ruby on Rails. ActiveRecord
- Проверка данных





- **БД (База данных)**
- **СУБД (Система управления базами данных)**
 - Сервер БД
 - Клиент БД
 - Библиотеки доступа к серверу БД
 - Средства обслуживания БД

СУБД

Модели данных



20.11.2018

- Реляционные
- Объектные
- Не реляционные
 - управляемые SQL
 - noSQL
 - Иерархические
 - XML/xPath/xQuery
 - JSON
 - Управляемые прочими языками

Реляционная модель

Эдгар Кодд (Edgar Codd)



20.11.2018

- n -арным отношением R называют подмножество декартова произведения множеств D_1, D_2, \dots, D_n ($n \geq 1$), не обязательно различных. Множества D называют доменами.
- Предусмотрены операции:
 - Объединение, Пересечение, Вычитание, Проекция, Декартово произведение, Выборка, Соединение, Деление.

Пример отношения



20.11.2018

- $D_1 = \{\text{Иванов, Петров, Сидоров}\}$
- $D_2 = \{\text{Физика, Химия}\}$
- $D_3 = \{3, 4, 5\}$

- $D_1 \rightarrow$ Фамилия
- $D_2 \rightarrow$ Предмет
- $D_3 \rightarrow$ Оценка

- столбцы (поля, атрибуты)
- строки (кортежи)
- число строк – мощность отношения

R_5		
Фамилия	Предмет	Оценка
Иванов	Физика	4
Иванов	Химия	3
Петров	Химия	5
Сидоров	Физика	5
Сидоров	Химия	4



- К. Дж. Дейт.
 - Введение в системы баз данных
 - Основы будущих систем баз данных: третий манифест
- Реляционная модель не требует **только простых типов!**
- Отношения могут быть вложенными.

Объектно-реляционное преобразование



20.11.2018

- Отображение объектов в терминах языка программирования на реляционную схему данных

Active Record

Общие положения



20.11.2018

- Соглашение об именах: Active Record
- класс Client отображается в client**s**
- Каждый объект Active Record – строка в таблице
- Поле id является первичным ключом

Параметры приложения, влияющие на имя таблиц



20.11.2018

- `config/application.rb`:
 - `config.active_record.pluralize_table_names = false`
 - `primary_key_prefix_type`
 - `table_name_prefix`
 - `table_name_suffix`

CRUD:

Creating, Reading, Updating, Deleting



20.11.2018

- Создание

```
c = Client.new
```

```
=> #<Client id: nil, name: nil, code: nil>
```

```
c.new_record?
```

```
=> true
```

```
c.persisted?
```

```
=> false
```

```
c = Client.new do |client|
```

```
  client.name = "Nile River Co."
```

```
  client.code = "NRC"
```

```
end
```

```
=> #<Client id: 1, name: "Nile River Co.", code: "NRC">
```

```
c = Client.create(:name => "Nile River, Co.", :code => "NRC")
```

```
=> #<Client id: 1, name: "Nile River, Co.", code: "NRC" ...>
```

Reading: динамические методы поиска по условию



20.11.2018

- Выборка данных

User.**find_by**(name: 'Spartacus', rating: 4)

City.find_by_**name**("Hackensack")

=> #<City id: 15942, name: "Hackensack", latitude: "40.8858330000", longitude: "-74.0438890000", state: "NJ">

Payment.find_by_**purchaser_and_state_and_country**

Person.where("user_name = ? AND password = ?", user_name, password)

Person.find_by_**user_name_and_password**(user_name, password)

- Сортировка

Payment.order("created_on").find_all_by_amount(50)

Явные SQL-запросы



20.11.2018

- Простой запрос

```
Client.find_by_sql("select * from clients")
```

```
=> [#<Client id: 1, name: "Paper Jam Printers", code: "PJP" ...>,  
#<Client id: 2, name: "Goodness Steaks", code:  
  "GOOD_STEAKS" ...>]
```

```
Client.find_by_sql("select * from clients where code like 'A%")
```

```
=> [#<Client id: 1, name: "Amazon, Inc" ...>]
```

- Параметризованный запрос

```
param = "A"
```

```
Client.where("code like ?", "#{param}%")
```

```
=> [#<Client id: 1, name: "Amazon, Inc" ...>]
```

SQL-инъекции



20.11.2018

```
Client.where("code like '#{params[:code]}%")  
=> [#<Client id: 1, name: "Amazon, Inc" ...>] # NOOOOOO!
```

Вставка **"Amazon"; DELETE FROM users;"**
=> полное уничтожение!

- Решение – параметризованные запросы

Rails ≤3 specifics



20.11.2018

Rails 3

Post.all.class #=> **Array**

Rails 4

Post.all.class #=> ActiveRecord::**Relation**::ActiveRecord_Relation_Posts

- Модифицирующие методы

posts = Post.all

Rails 3:

posts = posts.where(title: 'Sample')

Rails 4

posts.where! (title: 'Sample')

НОВЫЕ ВОЗМОЖНОСТИ Rails 5



20.11.2018

- `User.where(name: 'Jon').or(User.where(name: 'John'))`
instead of
`User.where('name = ? OR ?', 'Jon', 'John')`
- JSON support for ActiveRecord
- ...
- http://edgeguides.rubyonrails.org/5_0_release_notes.html#active-record
- <http://www.notch8.com/rails-5-activerecord-changes/>

Определение количества записей



20.11.2018

Client.count

```
# SELECT count(*) AS count_all FROM clients
```

Client.where(first_name: 'Ryan').count

```
# SELECT count(*) AS count_all FROM clients WHERE  
  (first_name = 'Ryan')
```

Client.count_by_sql("select count(*) from clients")

=> 132

Updating Обновление



20.11.2018

```
class ProjectController < ApplicationController
  def update
    Project.update(params[:id], params[:project])
    redirect_to projects_path
  end
  def mass_update
    Project.update(params[:projects].keys, params[:projects].values)
    redirect_to projects_path
  end
end
```



- Обновление одного экземпляра, найденного по первичному ключу

```
project = Project.find(1)
project.manager = 'Brett M.'
project.save
=> true
```

- Обновление с проверкой результата

```
class StoryController < ApplicationController
  def points
    story = Story.find(params[:id])
    if story.update(:points, params[:value])
      render :text => "#{story.name} updated"
    else
      render :text => "Error updating story points"
    end
  end
end
```

Обновление с проверкой результата



20.11.2018

```
class ProjectController < ApplicationController
  def update
    project = Project.find(params[:id])
    if project.update(params[:project])
      redirect_to project
    else
      render 'edit'
    end
  end
end
```

Обновление всех записей



20.11.2018

Обновить всех клиентов с указанными атрибутами

```
Customer.update_all wants_email: true
```

Обновить все книги с заголовком 'Rails'

```
Book.where('title LIKE ?', '%Rails%').update_all(author: 'David')
```

Обновить по условию 5 наиболее старых книг

```
Book.where('title LIKE ?', '%Rails%').order(:created_at).limit(5).  
  update_all(author: 'David')
```

- `update_all` — формирует SQL запрос
 - быстрый
 - исключает механизмы Active Record `validates` и `callbacks`
 - не обновляет поля типа `updated_at`
- https://apidock.com/rails/ActiveRecord/Relation/update_all

Touching Records

Изменение даты модификации



20.11.2018

```
user = User.first
```

```
user.touch #=> sets updated_at to now.
```

```
user.touch(:viewed_at) # sets viewed_at and  
updated_at to now.
```

```
class User < ActiveRecord::Base
```

```
  belongs_to :client, :touch => true
```

```
end
```

```
user.touch #=> also calls user.client.touch
```

Манипулирование атрибутами до преобразования



20.11.2018

```
class Timesheet < ActiveRecord::Base
  before_validation :fix_rate
  def fix_rate
    self[:rate] = rate_before_type_cast.tr('$','')
  end
end
```

Ограничение доступа к атрибутам



20.11.2018

```
class Customer < ActiveRecord::Base
  attr_protected :credit_rating
end
```

```
customer = Customer.new(:name => "Abe",
                        :credit_rating => "Excellent")
```

```
customer.credit_rating # => nil
```

```
customer.attributes = { "credit_rating" => "Excellent" }
```

```
customer.credit_rating # => nil
```

and now, the allowed way to set a credit_rating

```
customer.credit_rating = "Average"
```

```
customer.credit_rating # => "Average"
```

Атрибуты только для чтения



20.11.2018

```
class Customer < ActiveRecord::Base
  attr_readonly :social_security_number
end
```

```
customer = Customer.new(:social_security_number => "130803020")
=> #<Customer id: 1, social_security_number: "130803020", ...>
customer.social_security_number
=> "130803020"
customer.save
customer.social_security_number = "0000000000" # Note, no error raised!
customer.social_security_number
=> "0000000000"
customer.save
customer.reload
customer.social_security_number
=> "130803020" # the original readonly value is preserved
```


Удаление записей



20.11.2018

```
bad_timesheet = Timesheet.find(1)  
bad_timesheet.destroy
```

```
Timesheet.destroy([2, 3])
```

- delete использует SQL
 - Быстрее destroy
 - Игнорирует любые проверки и обработчики Active Records

```
Timesheet.delete(1)
```

Сложные условия where(conditions)



20.11.2018

```
Product.where(:sku => params[:sku])
```

```
Product.where(:sku => [9400,9500,9900])
```

```
Product.where(  
  'description like ? and color = ?',  
  "%#{terms}%", color)
```

```
Product.where('sku in (?)', selected_skus)
```

Связывание с переменными



20.11.2018

```
Product.where(  
  "name = :name AND sku = :sku AND created_at > :date",  
  :name => "Space Toilet",  
  :sku => 80800,  
  :date => '2009-01-01')
```

```
Message.where("subject LIKE :foo OR body LIKE :foo",  
  :foo => '%woah%')
```

```
User.where(:login => login, :password => password).first
```

```
Timesheet.where('submitted = ?', true)
```



20.11.2018

Порядок сортировки результата

```
Timesheet.order('created_at desc')  
Timesheet.order(:created_at)
```

- Случайный порядок

MariaDB (ранее MySQL)

```
Timesheet.order('RAND()')
```

Postgres

```
Timesheet.order('RANDOM()')
```

Microsoft SQL Server

```
Timesheet.order('NEWID()') # uses random uuids to sort
```

Oracle

```
Timesheet.order('dbms_random.value').first
```

Конструирование запроса



20.11.2018

```
def self.find_tagged_with(list)
  select("#{table_name}.*").
  from("#{table_name}, tags, taggings").
  where("#{table_name}.#{primary_key} =
    taggings.taggable_id and taggings.tag_id = tags.id and
    tags.name IN (?)",
    Tag.parse(list))
end
```

Миграции и правила их создания



20.11.2018

- DDL - Data Definition Language
- Код на Ruby для модификации БД
- rails generate migration
rails generate migration MigrationName
[field:type field:type] [options]
20181014090912_migration_name.rb

2018	10	14	09	09	12_...
год	месяц	день	часы	минуты	секунды

Migrations

create_table



20.11.2018

```
class CreateProducts < ActiveRecord::Migration
  def up
    create_table :products do |t|
      t.string :name
      t.text :description

      t.timestamps
    end
  end

  def down
    drop_table :products
  end
end
```

Migrations. Fix bad data

Исправление данных



20.11.2018

```
class AddReceiveNewsletterToUsers <
  ActiveRecord::Migration
  def up
    change_table :users do |t|
      t.boolean :receive_newsletter, :default => false
    end
    User.update_all ["receive_newsletter = ?", true]
  end

  def down
    remove_column :users, :receive_newsletter
  end
end
```


Migrations.

Метод change



20.11.2018

```
class CreateProducts < ActiveRecord::Migration
  def change
    create_table :products do |t|
      t.string :name
      t.text :description

      t.timestamps
    end
  end
end
```



- up - выполнить преобразование БД
- down – вернуть БД в состояние до миграции.

- add_column
- add_index
- change_column
- change_table
- create_table
- drop_table
- remove_column
- remove_index
- rename_column

Соответствие типов полей



20.11.2018

Migration Type	MariaDB	Postgres	SQLite	Oracle	Ruby Class
:binary	blob	bytea	blob	blob	String
:boolean	tinyint(1)	boolean	boolean	number(1)	Boolean
:date	date	date	date	date	Date
:datetime	datetime	timestamp	datetime	date	Time
:decimal	decimal	decimal	decimal	decimal	BigDecimal
:float	float	float	float	number	Float
:integer	int(11)	integer	integer	number(38)	Fixnum
:string	varchar(255)	character	varchar(255)	varchar2(255)	String
:text	text	clob(32768)	text	clob	String
:time	time	time	time	date	Time
:timestamp	datetime	timestamp	datetime	date	Time

Адаптеры СУБД



20.11.2018

- `Ruby193/lib/ruby/gems/2.x.x/gems/activerecord-x.x.x/lib\active_record/connection_adapters/`

postgresql_adapter.rb:

```
NATIVE_DATABASE_TYPES = {
  :primary_key => "serial primary key".freeze,
  :string => { :name => "character varying", :limit => 255 },
  :text => { :name => "text" },
  :integer => { :name => "integer" },
  :float => { :name => "float" },
  :decimal => { :name => "decimal" },
  :datetime => { :name => "timestamp" },
  :timestamp => { :name => "timestamp" },
  :time => { :name => "time" },
  :date => { :name => "date" },
  :binary => { :name => "bytea" },
  :boolean => { :name => "boolean" },
  :xml => { :name => "xml" }
}
```

Опции полей (Column Modifiers)



20.11.2018

```
class AddDetailsToProducts < ActiveRecord::Migration
  def change
    add_column :products, :price, :decimal, precision: 5, scale: 2
  end
end
```

- :default => value
- :limit => size
- :null => false
- :precision => number
- :scale => number
- <http://guides.rubyonrails.org/migrations.html#column-modifiers>

Запуск миграций



20.11.2018

- **rake db:migrate VERSION=20180906120000**
- **rake db:rollback** – откат предыдущей
- **rake db:rollback STEP=3** – откат последних трёх
- **rake db:migrate:redo STEP=3** – откат и повторное применение
- **rake db:migrate:up VERSION=20180906120000** – запуск указанного метода

Генерация модели



20.11.2018

- rails generate model Product name:string description:text

- Создаются модель + миграция

```
class CreateProducts < ActiveRecord::Migration
  def change
    create_table :products do |t|
      t.string :name
      t.text :description

      t.timestamps
    end
  end
end
```

Генерация только миграции



20.11.2018

- rails generate migration AddPartNumberToProducts

```
class AddPartNumberToProducts < ActiveRecord::Migration
  def change
  end
end
```


Генерация миграции



20.11.2018

- rails generate migration AddPartNumberToProducts
part_number:string

```
class AddPartNumberToProducts <
  ActiveRecord::Migration
  def change
    add_column :products, :part_number, :string
  end
end
```



20.11.2018

Распознавание имени миграции

- rails generate migration **Remove**PartNumberFrom**Products**
part_number:string

```
class RemovePartNumberFromProducts < ActiveRecord::Migration
  def up
    remove_column :products, :part_number
  end

  def down
    add_column :products, :part_number, :string
  end
end
```

Распознавание имени миграции



20.11.2018

- rails generate migration **AddDetailsToProducts**
part_number:string price:decimal

```
class AddDetailsToProducts < ActiveRecord::Migration
  def change
    add_column :products, :part_number, :string
    add_column :products, :price, :decimal
  end
end
```

Associations



20.11.2018

- Связи между сущностями

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

books	
Model: Book belongs_to :author	
id	integer
author_id	integer
published_at	datetime



authors	
Model: Author	
id	integer
name	string

```
class Book < ActiveRecord::Base
  belongs_to :author
end
```

- http://guides.rubyonrails.org/association_basics.html

One-to-Many Relationships



20.11.2018

```
class User < ActiveRecord::Base
  has_many :timesheets
  has_many :expense_reports
end
```

```
class Timesheet < ActiveRecord::Base
  belongs_to :user
end
```

```
class ExpenseReport < ActiveRecord::Base
  belongs_to :user
end
```

```
User.limit(100).map(&:timesheets) # 1 + 100 SQL запросов
User.limit(100).includes(:timesheets).map(&:timesheets) # 2 SQL запроса
```

- http://guides.rubyonrails.org/association_basics.html
- <http://tomdallimore.com/blog/includes-vs-joins-in-rails-when-and-where/>

Database Seeding

Начальное заполнение БД



20.11.2018

- db/seeds.rb

```
User.create!( :login => 'admin',  
              :email => 'admin@tr3w.com',  
              :password => '123', :password_confirmation => '123',  
              :authorized_approver => true)  
client = Client.create!( :name => 'Workbeast', :code => 'BEAST')  
client.billing_codes.create!( :name => 'Meetings', :code => 'MTG')  
client.billing_codes.create!( :name => 'Development', :code => 'DEV')
```

```
c = Client.find_or_create_by_name_and_code!('Workbeast', 'BEAST')  
c.billing_codes.find_or_create_by_name_and_code!('Meetings', 'MTG')  
c.billing_codes.find_or_create_by_name_and_code!('Development', 'DEV')
```



- rake db:create
- db:drop
- db:forward and db:rollback
- db:migrate
- db:schema:dump
 - Формирует файл db/schema.rb
- db:seed
- db:structure:dump

Дополнительные возможности ActiveRecord



20.11.2018



- **validates_acceptance_of**

```
class Account < ActiveRecord::Base
  validates_acceptance_of :privacy_policy, :terms_of_service
end
```

```
class Cancellation < ActiveRecord::Base
  validates_acceptance_of :account_cancellation, :accept => 'YES'
end
```

validates_confirmation_of



20.11.2018

```
class Account < ActiveRecord::Base
  validates_confirmation_of :password
end
```

```
= form_for account do |f|
  = f.label :login
  = f.text_field :login
  = f.label :password
  = f.password_field :password
  = f.label :password_confirmation
  = f.password_field :password_confirmation
  = f.submit
```

validates_each



20.11.2018

- Проверка указанных атрибутов с пользовательской функцией

```
class Invoice < ActiveRecord::Base
  validates_each :supplier_id, :purchase_order do |record, attr, value|
    record.errors.add(attr) unless PurchasingSystem.validate(attr, value)
  end
end
```

validates_format_of



20.11.2018

```
class Person < ActiveRecord::Base
  validates_format_of :email,
    :with => /\A([^\s]+)@((?:[-a-z0-9]+\.)+[a-z]{2,})\Z/i
end

validates_format_of :name, :with =>
  /^((localhost)|#{DOMAIN}|#{NUMERIC_IP})#{PORT}$/
```



20.11.2018

validates_inclusion_of

```
class Person < ActiveRecord::Base
  validates_inclusion_of :gender,
                        :in => ['m','f'],
                        :message => 'ORLY?'
end
```

...

```
class Account < ActiveRecord::Base
  validates_exclusion_of :login,
                       :in => ['admin', 'root', 'superuser'],
                       :message => 'Naughty, naughty!'
end
```

validates_length_of



20.11.2018

```
class Account < ActiveRecord::Base
  validates_length_of :login, :minimum => 5
end
```

```
class Account < ActiveRecord::Base
  validates_length_of :login, :within => 5..20
end
```

validates_presence_of validates_uniqueness_of



20.11.2018

```
class Account < ActiveRecord::Base
  validates_presence_of :login, :email, :account_number
end
```

```
class Account < ActiveRecord::Base
  validates_uniqueness_of :login
end
```



- Регистрация в классе модели

```
class Beethoven < ActiveRecord::Base
  before_destroy :last_words
  protected
  def last_words
    logger.info "Plaudite, amici, comedia finita est"
  end
end
```

- <http://api.rubyonrails.org/classes/ActiveRecord/Callbacks.html>

Observers



20.11.2018

- Вызываются **ПОСЛЕ** выполнения действия
- Размещаются в отдельных файлах
- Часто используются для исправления данных в записях
- Удалены из **Rails 4**, но могут использоваться: `gem 'rails-observers'`

```
class AccountObserver < ActiveRecord::Observer
  def after_create(model)
    DEFAULT_AUDIT_LOG.created(model.inspect)
  end
  def after_update(model)
    DEFAULT_AUDIT_LOG.updated(model.inspect)
  end
  def after_destroy(model)
    DEFAULT_AUDIT_LOG.destroyed(model.inspect)
  end
end
```

Concerns



20.11.2018

```
module MyConcernModule
  extend ActiveSupport::Concern

  included do
    after_save :do_something
  end
  def do_something
    ...
  end
end

class MyModel < ActiveRecord::Base
  include MyConcernModule
end
```

- <http://api.rubyonrails.org/classes/ActiveSupport/Concern.html>
- <http://stackoverflow.com/questions/15165260/rails-observer-alternatives-for-4-0>

Пример создания приложения



20.11.2018

- rails new CalcFunc
- cd CalcFunc
- rails generate controller Calc index view
- rails generate model CalcResult from:float to:float
step:float result:text

Созданная миграция



20.11.2018

- db/migrate/20181025165857_create_calc_results.rb

```
class CreateCalcResults < ActiveRecord::Migration
  def change
    create_table :calc_results do |t|
      t.float :from
      t.float :to
      t.float :step
      t.text :result

      t.timestamps
    end
  end
end
```

Контроллер Calc



20.11.2018

```
class CalcController < ApplicationController
  def index
  end
  def view
    from, to, step = params[:from].to_f, params[:to].to_f, params[:step].to_f

    if res = CalcResult.find_by_from_and_to_and_step( from, to, step )
      @result = ActiveSupport::JSON::decode( res.result )
    else
      @result = from.step(to, step).collect{ |x| [x, Math.sin(x)] }
      res = CalcResult.create :from => from, :to => to, :step => step,
                             :result => ActiveSupport::JSON::encode( @result )

      res.save
    end
  end
  def results
    @result = CalcResult.all
    render xml: @result
  end
end
```



```
<%= t(:'Sin calculation') %>
```

```
<%= form_tag("/calc/view", :method => "get") do %>
```

```
  <%= label_tag(t(:'From:')) %>
```

```
  <%= text_field_tag(:from) %> <br/>
```

```
  <%= label_tag(t(:'To:')) %>
```

```
  <%= text_field_tag(:to) %> <br/>
```

```
  <%= label_tag(t(:'Step:')) %>
```

```
  <%= text_field_tag(:step) %> <br/>
```

```
  <%= submit_tag(t(:'View result')) %>
```

```
<% end %>
```

view.html.erb



20.11.2018

```
<table>
<tr>
  <td>x</td>
  <td>val</td>
</tr>
<% @result.each do |item| %>
  <tr>
    <td><%= item[0] %></td>
    <td><%= item[1] %></td>
  </tr>
<% end %>
</table>

<%= link_to "Repeat calculation", :calc_index %>
```

Локализация Rails-приложения



20.11.2018

- locales/ru.yml (**UTF-8, отступы обязательны!**)

ru:

'Sin calculation': 'Вычисление синуса'

'From:': 'От:'

'To:': 'До:'

'Step:': 'Шаг:'

'View result': 'Просмотреть результат'

- config/application.rb

config.i18n.default_locale = :ru

- <http://guides.rubyonrails.org/i18n.html>



- <http://guides.rubyonrails.org/migrations.html>
- http://guides.rubyonrails.org/active_record_querying.html
- http://guides.rubyonrails.org/association_basics.html
- Obie Fernandez. The Rails 5 Way. Addison-Wesley. 2017
- <http://rusrails.ru/>
- <https://www.railstutorial.org/book>