

PATS Models

```
class Owner < ApplicationRecord
  has_many :pets
  has_many :visits, through: :pets

  scope :alphabetical, -> { order('last_name, first_name') }
  scope :active, -> { where(active: true) }
  scope :inactive, -> { where.not(active: true) }
  scope :search, ->(term){ where('first_name LIKE ? OR last_name LIKE ?', "#{term}%",
                                "#{term}%")}
end

class Animal < ApplicationRecord
  has_many :pets
  has_many :animal_medicines
  has_many :medicines, through: :animal_medicines

  scope :alphabetical, -> { order('name') }
  scope :active, -> { where(active: true) }
  scope :inactive, -> { where(active: false) }
end

class Medicine < ApplicationRecord
  has_many :animal_medicines
  has_many :animals, :through => :animal_medicines
  has_many :medicine_costs
  has_many :dosages
  has_many :visits, :through => :dosages

  scope :alphabetical, -> { order('name') }
  scope :active, -> { where(active: true) }
  scope :depleted, -> { where('stock_amount < ?', 100) }
  scope :vaccines, -> { where(vaccine: true) }
  scope :nonvaccines, -> { where(vaccine: false) }
  scope :for_animal, ->(animal_id){
    joins(:animal_medicines).where('animal_medicines.animal_id = ?', animal_id) }
end

class AnimalMedicine < ApplicationRecord
  belongs_to :animal
  belongs_to :medicine

  scope :for_animal, ->(animal_id) { where(animal_id: animal_id) }
  scope :for_medicine, ->(medicine_id) { where(medicine_id: medicine_id) }
end

class Procedure < ApplicationRecord
  has_many :procedure_costs
  has_many :treatments
  has_many :visits, through: :treatments

  scope :alphabetical, -> { order('name') }
  scope :active, -> { where(active: true) }
  scope :inactive, -> { where.not(active: true) }
end
```

```
class MedicineCost < ApplicationRecord
  belongs_to :medicine

  scope :chronological, -> { order('start_date') }
  scope :current, -> { where("end_date IS NULL") }
  scope :for_date, ->(date){ where("start_date <= ? AND (end_date > ? OR end_date IS NULL)",
                                date, date) }
  scope :for_medicine, ->(medicine_id) { where(medicine_id: medicine_id) }
end

class ProcedureCost < ApplicationRecord
  belongs_to :procedure

  scope :chronological, -> { order('start_date') }
  scope :current, -> { where(end_date: nil) }
  scope :for_date, ->(date) { where("start_date <= ? AND (end_date > ? OR end_date IS NULL)",
                                date, date) }
  scope :for_procedure, ->(procedure_id) { where(procedure_id: procedure_id) }
end

class Pet < ApplicationRecord
  belongs_to :animal
  belongs_to :owner
  has_many :visits
  has_many :dosages, through: :visits
  has_many :treatments, through: :visits

  scope :alphabetical, -> { order('name') }
  scope :active, -> { where(active: true) }
  scope :inactive, -> { where.not(active: true) }
  scope :females, -> { where(female: true) }
  scope :males, -> { where(female: false) }
  scope :for_owner, ->(owner_id) { where("owner_id = ?", owner_id) }
  scope :by_animal, ->(animal_id) { where("animal_id = ?", animal_id) }
  scope :born_before, ->(dob) { where('date_of_birth < ?', dob) }
end

class Visit < ApplicationRecord
  belongs_to :pet
  has_one :animal, through: :pet
  has_many :dosages
  has_many :treatments
  has_one :owner, through: :pet

  scope :chronological, lambda { order('date DESC') }
  scope :for_pet, ->(pet_id) { where('pet_id = ?', pet_id) }
  scope :last_x, ->(num) { limit(num).order('date DESC') }
end

class Dosage < ApplicationRecord
  belongs_to :medicine
  belongs_to :visit
  has_one :pet, through: :visit

  scope :for_medicine, ->(medicine_id) { where(medicine_id: medicine_id) }
  scope :for_visit, ->(visit_id) { where(visit_id: visit_id) }
end
```

```
class Treatment < ApplicationRecord
  belongs_to :visit
  belongs_to :procedure
  has_one :pet, :through => :visit

  scope :for_procedure, ->(procedure_id) { where("procedure_id = ?", procedure_id) }
  scope :for_visit, ->(visit_id) { where("visit_id = ?", visit_id) }
  scope :alphabetical, -> { joins(:procedure).order("procedures.name") }
end
```