МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Кафедра програмних засобів

Звіт

з лабораторної роботи №9

з дисципліни: «Крос-платформне програмування»

за темою: «RUBY ON RAILS. ЗВ’ЯЗОК МІЖ МОДЕЛЯМИ. СТВОРЕННЯ МІКРО ПОВІДОМЛЕНЬ. MVC ШАБЛОН ПРОЕКТУВАННЯ. ЗАЛУЧЕННЯ СИСТЕМИ КОНТРОЛЮ ВЕРСІЙ GIT ДО ПРОЕКТІВ НА RUBY ON RAILS»

Виконав:

ст. гр. КНТ-117 М. І. Басанець

Прийняв:

старший викладач Є. М. Федорченко

2020

1 МЕТА РОБОТИ

Навчитися зв’язувати дані в Ruby on Rails. Навчитися працювати з шаблоном MVC. Навчитися працювати з Git.

2 зАВДАННЯ ДО РОБОТИ

Розробити модель Держава, модель Місто, модель визначна пам'ятка й кожну модель зв'язати один з одним через ід. Заповнити кожну таблицю як мінімум 3 записами Зв'язок продемонструвати через консоль.

3 Результати виконання завдання

Код файлу city.rb:

class City < ApplicationRecord

belongs\_to :country

has\_many :sights, dependent: :destroy

end

Код файлу country.rb:

class Country < ApplicationRecord

has\_many :cities, dependent: :destroy

end

Код файлу sight.rb:

class Sight < ApplicationRecord

belongs\_to :city

end

Код файлу cities\_controller.rb:

class CitiesController < ApplicationController

before\_action :set\_city, only: [:show, :edit, :update, :destroy]

# GET /cities

# GET /cities.json

def index

@cities = City.all

end

# GET /cities/1

# GET /cities/1.json

def show

end

# GET /cities/new

def new

@city = City.new

end

# GET /cities/1/edit

def edit

end

# POST /cities

# POST /cities.json

def create

@city = City.new(city\_params)

respond\_to do |format|

if @city.save

format.html { redirect\_to @city, notice: 'City was successfully created.' }

format.json { render :show, status: :created, location: @city }

else

format.html { render :new }

format.json { render json: @city.errors, status: :unprocessable\_entity }

end

end

end

# PATCH/PUT /cities/1

# PATCH/PUT /cities/1.json

def update

respond\_to do |format|

if @city.update(city\_params)

format.html { redirect\_to @city, notice: 'City was successfully updated.' }

format.json { render :show, status: :ok, location: @city }

else

format.html { render :edit }

format.json { render json: @city.errors, status: :unprocessable\_entity }

end

end

end

# DELETE /cities/1

# DELETE /cities/1.json

def destroy

@city.destroy

respond\_to do |format|

format.html { redirect\_to cities\_url, notice: 'City was successfully destroyed.' }

format.json { head :no\_content }

end

end

private

# Use callbacks to share common setup or constraints between actions.

def set\_city

@city = City.find(params[:id])

end

# Only allow a list of trusted parameters through.

def city\_params

params.require(:city).permit(:name, :population, :country\_id)

end

end

Код файлу countries\_controller.rb:

class CountriesController < ApplicationController

before\_action :set\_country, only: [:show, :edit, :update, :destroy]

# GET /countries

# GET /countries.json

def index

@countries = Country.all

end

# GET /countries/1

# GET /countries/1.json

def show

end

# GET /countries/new

def new

@country = Country.new

end

# GET /countries/1/edit

def edit

end

# POST /countries

# POST /countries.json

def create

@country = Country.new(country\_params)

respond\_to do |format|

if @country.save

format.html { redirect\_to @country, notice: 'Country was successfully created.' }

format.json { render :show, status: :created, location: @country }

else

format.html { render :new }

format.json { render json: @country.errors, status: :unprocessable\_entity }

end

end

end

# PATCH/PUT /countries/1

# PATCH/PUT /countries/1.json

def update

respond\_to do |format|

if @country.update(country\_params)

format.html { redirect\_to @country, notice: 'Country was successfully updated.' }

format.json { render :show, status: :ok, location: @country }

else

format.html { render :edit }

format.json { render json: @country.errors, status: :unprocessable\_entity }

end

end

end

# DELETE /countries/1

# DELETE /countries/1.json

def destroy

@country.destroy

respond\_to do |format|

format.html { redirect\_to countries\_url, notice: 'Country was successfully destroyed.' }

format.json { head :no\_content }

end

end

private

# Use callbacks to share common setup or constraints between actions.

def set\_country

@country = Country.find(params[:id])

end

# Only allow a list of trusted parameters through.

def country\_params

params.require(:country).permit(:name, :primary\_language, :area, :population)

end

end

Код файлу sights\_controller.rb:

class SightsController < ApplicationController

before\_action :set\_sight, only: [:show, :edit, :update, :destroy]

# GET /sights

# GET /sights.json

def index

@sights = Sight.all

end

# GET /sights/1

# GET /sights/1.json

def show

end

# GET /sights/new

def new

@sight = Sight.new

end

# GET /sights/1/edit

def edit

end

# POST /sights

# POST /sights.json

def create

@sight = Sight.new(sight\_params)

respond\_to do |format|

if @sight.save

format.html { redirect\_to @sight, notice: 'Sight was successfully created.' }

format.json { render :show, status: :created, location: @sight }

else

format.html { render :new }

format.json { render json: @sight.errors, status: :unprocessable\_entity }

end

end

end

# PATCH/PUT /sights/1

# PATCH/PUT /sights/1.json

def update

respond\_to do |format|

if @sight.update(sight\_params)

format.html { redirect\_to @sight, notice: 'Sight was successfully updated.' }

format.json { render :show, status: :ok, location: @sight }

else

format.html { render :edit }

format.json { render json: @sight.errors, status: :unprocessable\_entity }

end

end

end

# DELETE /sights/1

# DELETE /sights/1.json

def destroy

@sight.destroy

respond\_to do |format|

format.html { redirect\_to sights\_url, notice: 'Sight was successfully destroyed.' }

format.json { head :no\_content }

end

end

private

# Use callbacks to share common setup or constraints between actions.

def set\_sight

@sight = Sight.find(params[:id])

end

# Only allow a list of trusted parameters through.

def sight\_params

params.require(:sight).permit(:name, :city\_id)

end

end

Результати додавання записів через консоль зображено на рисунку 3.1.

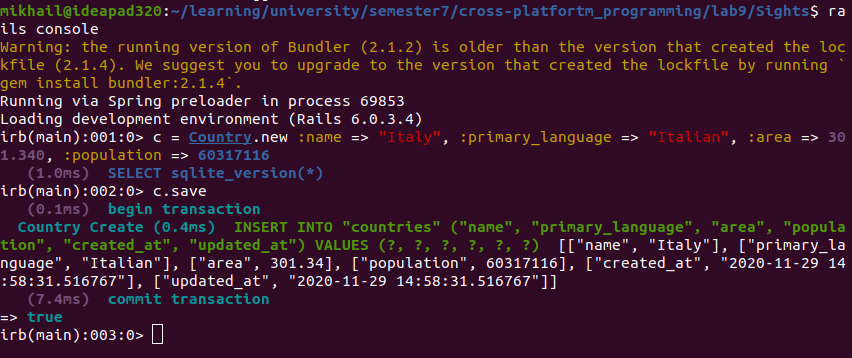


Рисунок 3.1 – Результати додавання записів через консоль

Перевірка заповненності таблиць через консоль зображена на рисунку 3.2.

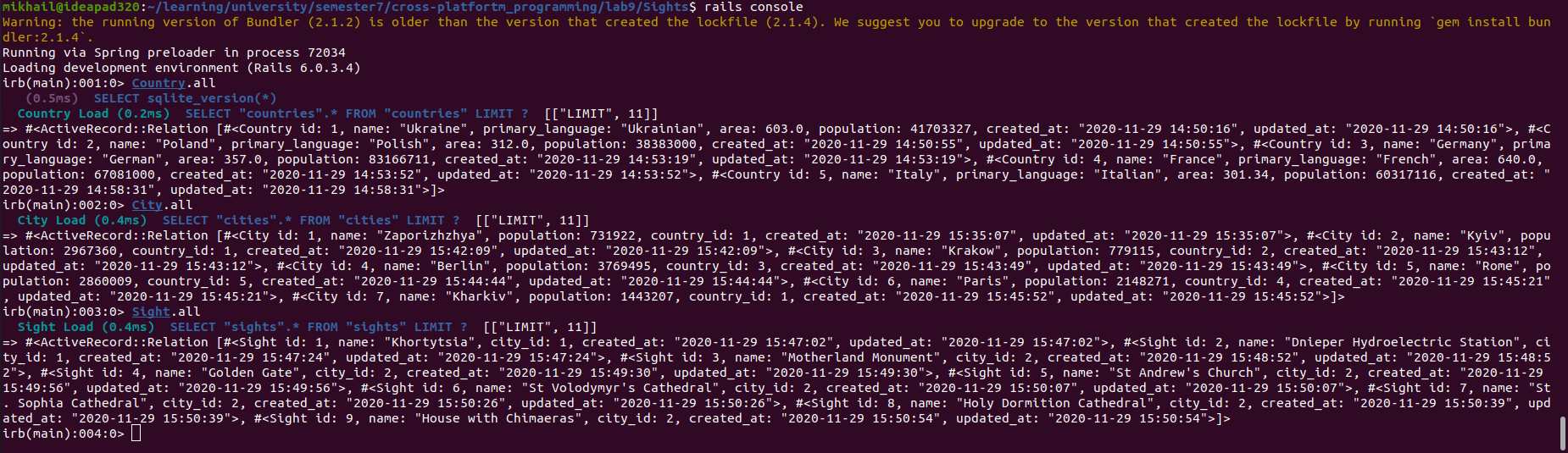


Рисунок 3.2 – Перевірка заповненності таблиць

Перевірка зв’язку між таблицями через консоль зображена на рисунку 3.3.

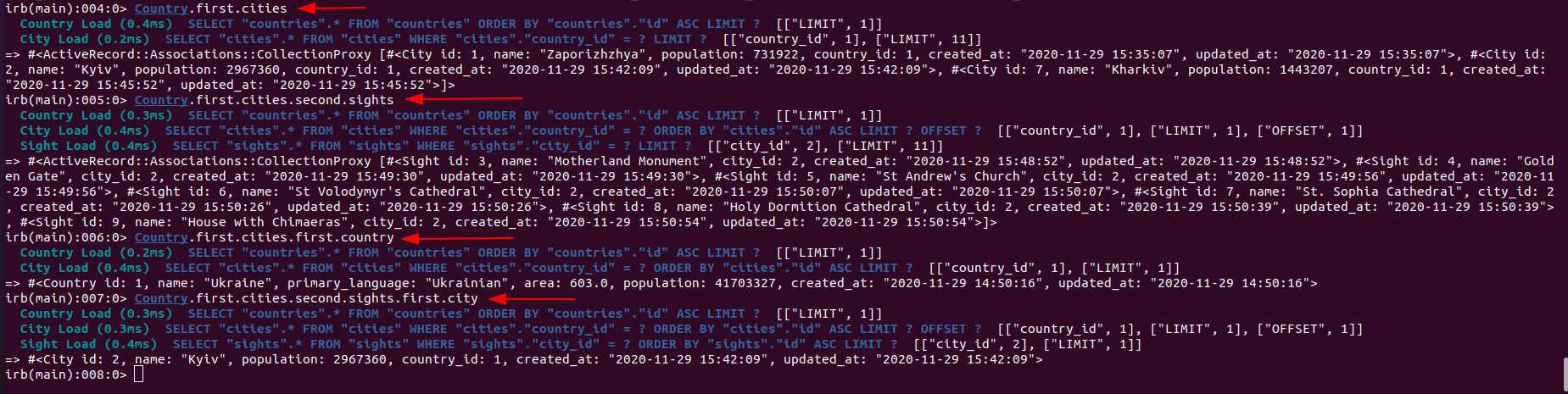


Рисунок 3.3 – Перевірка зв’язку таблиць

4 висновки

При виконанні даної лабораторної роботи на практиці засвоїв основні принципи та засади зв’язування даних в Ruby on Rails. Навчився працювати з шаблоном MVC.

Відповіді на контрольні запитання:

1. Що таке Scaffold?

Scaffold - швидкий спосіб генерації необхідних компонентів у Rails. Scaffold генерує модель, міграцію бази даних для моделі, контролер та відображення.

2. Що таке ApplicationRecord?

Active Record - це M у MVC - модель - яка є слоем в системі, відповідальною за представлення бізнес-логіки та даних. Active Record використовує створення та використання бізнес-об'єктів, дані яких вимагають постійного зберігання у базах даних.

3. Що таке MVC?

MVC - архітектурний шаблон. Цей шаблон передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

4. Які переваги при використанні Scaffold в Ruby?

Основна перевага - швидке створення каркасу проекту.

5. Які недоліки при використанні Scaffold в Ruby?

Основний недолік - відсутність гнучкості при створенні каркасу, оскільки великі проекти потребують більш специфічних налаштувань.