



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 Прикладная информатика

О Т Ч Е Т

по лабораторной работе № 1 1

Название: Добавление модели. ORM. Разработка БД, подключение,
хранение и поиск данных.

Дисциплина: Языки интернет-программирования

Студент

ИУ6-35 Б
(Группа)

08.12.2023

(Подпись, дата)

И.А. Дулина

(И.О. Фамилия)

Преподаватель

Е.Ю. Гаврилова

(Подпись, дата)

(И.О. Фамилия)

Вариант 8

Цель работы: освоить принципы применения средств объектно-реляционного преобразования в составе Ruby on Rails.

Задание:

Модифицировать код ЛР 8 таким образом, чтобы запросы, которые были ранее выполнены, сохранялись в БД и при следующем запросе не требовали повтора вычислений.

- Сформировать модель в соответствии с потребностями хранения данных. Входные параметры являются ключами, по которым извлекается результат.

- Выполнить создание БД и миграцию соответствующими запросами rake.

- Написать тест на добавление и поиск данных с помощью модели.

Проверить выполнение теста.

- Модифицировать код приложения таким образом, чтобы результат вычислений преобразовывался в строковый или бинарный формат (на выбор: json, xml, и пр.). Проверить через отладочную печать в консоль, что преобразование выполняется корректно.

- Вставить код для сохранения данных в БД и запрос на поиск предыдущего результата вычислений.

- Добавить действие в контроллер, позволяющее определить, что хранится в БД через сериализацию в XML.

- Проверить, что при выполнении запроса, данные добавляются в БД.

- При помощи консоли сообщений Puma/Webrick определить, производится ли поиск результата предыдущего запроса в БД и не повторяются ли одни и те же вычисления.

- Модифицировать модель таким образом, чтобы добавление записей с одинаковыми параметрами было невозможно (string:text:uniq)

- Реализовать тест модели, проверяющий невозможность повторного добавления одних и тех же результатов вычислений.

- Реализовать функциональный тест, проверяющий, что результаты вычислений различны при различных входных параметрах.

- Проверить маршруты приложения с помощью rake routes и убрать лишние.
- Обеспечить доступ при обращении по адресу /.

В консоли:

```
$rails new chisla11
```

```
$cd chisla11
```

```
$rails generate controller Chisla input view
```

Сформируем модель (с невозможностью добавления одинаковых параметров, т.е. параметр string:text:uniq):

```
$rails g scaffold ChislaResult string:text:uniq result:text my_table:json
```

Создание бд:

```
$rake db:create
```

Создание миграции:

```
$rake db:migrate
```

В gem file добавляем строку

```
gem 'rails-controller-testing'
```

config/routes.rb добавить

```
root 'chisla#input'
```

db/migrate/20231120135713_create_chisla_results.rb

```
class CreateChislaResults < ActiveRecord::Migration[7.0]
```

```
  def change
```

```
    create_table :chisla_results do |t|
```

```
      t.text :string
```

```
      t.text :result
```

```
      t.json :my_table
```

```
      t.timestamps
```

```
    end
```

```
    add_index :chisla_results, :string, unique: true
```

```
  end
```

```
end
```

app/views/chisla/input.html.erb

<h1>Chisla#input</h1>

<p>Find me in app/views/chisla/input.html.erb</p>

<div>

<form action="/chisla/view" method="get" data-remote="true" id="chisla_form"
accept-charset="UTF-8">

<div>

<label for="str">Введите не менее 10 чисел

<input type="text" id="str" name="str" required/>

</label>

</div>

<div>

<input type="submit" value="Найти наиболее длинную монотонно
возрастающую последовательность"/>

</div>

</form>

</div>

app/views/chisla/view.html.erb

<h1>Chisla#view</h1>

<p>Find me in app/views/chisla/view.html.erb</p>

<p>Таблица результатов</p>

<%= @result[1].html_safe %>

<%= link_to "Найти для других чисел", :chisla_input %>

app/controllers/chisla_controller.rb

require 'json'

class ChislaController < ApplicationController

def input

```

end

def view
  if params[:str]
    begin
      res = params[:str].scan(/-?\d+(?:\.\d+)?/).map(&:to_i)
      raise StandardError if res.length < 10
      @result = check(res)
    rescue StandardError
      @result = [{}, 'Что-то пошло не так']
    end
  else
    @result = [{}, 'Unknown!']
  end
end

def check(res)
  #поиск предыдущего результата вычислений
  if (res_bs = ChislaResult.find_by_string(res.join(' ')))
    returning = [ActiveSupport::JSON::decode(res_bs.result),
JSON.parse(res_bs.my_table)]
    pp 'Результат уже посчитан'

  else
    # сохранение данных в бд:
    returning = create(res)
    pp 'Результат просчитан впервые'
    res_bs = ChislaResult.create :string => res.join(' '), :result =>
ActiveSupport::JSON::encode(returning[0]), :my_table => returning[1].to_json
    res_bs.save
  end
end

```

```

    print 'Запись добавлена в БД ', res_bs, "\n"
end
sol = returning[0]
everything = returning[1]
[sol, create_table(everything)]
end

```

```

def create(res)
  max = 0
  all = []
  solution = ""
  i = 0
  loop do
    len = 1
    posl = [res[i]]
    i += 1
    while res[i] > res[i - 1]
      len += 1
      posl << res[i]
      break unless i + 1 < res.length

      i += 1
    end
    all << posl.join(' ')
    if len > max
      max = len
      solution = posl.join(' ')
    end
    break if i + 1 == res.length
  end
end

```

```

end
result = []
all.length.times do |j|
  str = if solution == all[j]
    '+'
  else
    ''
  end
  result << if j.zero?
    [res.join(' '), all[j], str]
  else
    [' ', all[j], str]
  end
end
[solution, result]
end

def create_table(result)
  rows = "<tr><th>#{'Изначальный'}</th><th>#{'Все возможные'}</th><th>#{'Самая длинная'}</th></tr>"
  result.each do |init, all, sol|
    rows += "<tr><td>#{init}</td><td>#{all}</td><td>#{sol}</td></tr>"
  end
  @table = "<table border='1' class='table'><tbody>#{rows}</tbody></table>"
end

end

app/controllers/chisla_results_controller.rb добавить
XSLT_TRANSFORM = "#{Rails.root}/public/some_transformer.xslt".freeze #
Путь до xslt файла

```

Добавить действие в контроллер, позволяющее определить, **что хранится в БД** через сериализацию в XML.

```
#http://127.0.0.1:3000/show_all.xml
def show_all
  respond_to do |format|
    results = ChislaResult.all
    rows = "
    results.each do |record|
      rows +=
"<cd><former>#{record.string}</former><res_string>#{record.result}</res_string><my_t>#{record.my_table}</my_t></cd>"
    end
    response = "<catalog>#{rows}</catalog>"
    format.xml { render xml: xslt_transform(response).to_xml }
  end
end
```

```
def xslt_transform(data, transform: XSLT_TRANSFORM)
  # Функция преобразования
  pp 'checkpoint2'
  print data, transform, "\n"
  doc = Nokogiri::XML(data)
  xslt = Nokogiri::XSLT(File.read(transform))
  xslt.transform(doc)
end
```

public/some_transformer.xslt

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
version="1.0"
```



```

<xsl:template match="/">
  <xsl:if test="output/input">
    <div><xsl:value-of select="output/input"/></div>
  </xsl:if>

  <table border="1">
    <tr bgcolor="#9933ff">
      <th>Former string</th>
      <th>Result</th>
      <th>Table</th>
    </tr>
    <xsl:for-each select="catalog/cd">
      <tr>
        <td><xsl:value-of select="former"/></td>
        <td><xsl:value-of select="res_string"/></td>
        <td><xsl:value-of select="my_t"/></td>
      </tr>
    </xsl:for-each>
  </table>
</xsl:template>
</xsl:stylesheet>

```

```
test/controllers/chisla_controller_test.rb
```

```
require "test_helper"
```

```
# rake test TEST=test/controllers/chisla_controller_test.rb - запуск теста
```

```
class ChislaControllerTest < ActionDispatch::IntegrationTest
```

```
  # rake db:migrate RAILS_ENV=test - запуск миграции для теста
```

```
  # Написать тест на добавление и поиск данных с помощью модели.
```

Проверить выполнение теста.

```

def add_record(str = '1 2 3 4 5 6 7 8 9 10', data = '1 2 3 4 5 6 7 8 9 10', table = '')
  record = ChislaResult.create :string => str, :result =>
ActiveSupport::JSON::encode(data), :my_table => table
  record.save
  pp 'Добавлено'
  record
end

```

```

test "Add data" do
  record = add_record('1 2 3 4 -1 -2 1 2 3 1', '1 2 3 4')
  assert record
end

```

```

test "Find data" do
  add_record
  record = ChislaResult.find_by_string('1 2 3 4 5 6 7 8 9 10')
  assert record
end

```

тест на невозможность повторения

```

test "Add same result" do
  add_record
  assert_raises(ActiveRecord::RecordNotUnique) do
    add_record
  end
end

```

проверка маршрутов: rails routes --expanded

```

test "should get input" do
  get chisla_input_url

```

```

    assert_response :success
  end

  test "should get view" do
    get chisla_view_url
    assert_response :success
  end

  # тест на то, что результаты вычислений различны при различных
  входных параметрах
  test "different" do
    get chisla_view_url, params: { str: '1 2 3 4 5 6 7 8 9 10' }
    result1 = assigns[:result]

    get chisla_view_url, params: { str: '-1 -2 -3 2 3 4 -2 -3 2 3' }
    result2 = assigns[:result]
    pp result2
    assert_not_same result1, result2
  end

  #тест на то, что запись уже добавлена
  test 'already added' do
    add_record('-1 -2 -3 2 3 4 -2 -3 2 3', '-3 2 3 4', '["-1 -2 -3 2 3 4 -2 -3 2 3", "-1", "
"], [" ", "-2", " "], [" ", "-3 2 3 4", "+"], [" ", "-2", " "], [" ", "-3 2 3", " "]]')
    get chisla_view_url, params: { str: '-1 -2 -3 2 3 4 -2 -3 2 3' }
    result2 = assigns[:result]
    pp result2
    assert_equal(true, result2[2])
  end
end

end

config/routes.rb
Rails.application.routes.draw do

```

```

resources :chisla_results
get 'show_all', to: 'chisla_results#show_all'
get 'chisla/input'
get 'chisla/view'
# Define your application routes per the DSL in
https://guides.rubyonrails.org/routing.html
root 'chisla#input'
# Defines the root path route ("/")
# root "articles#index"
end

```

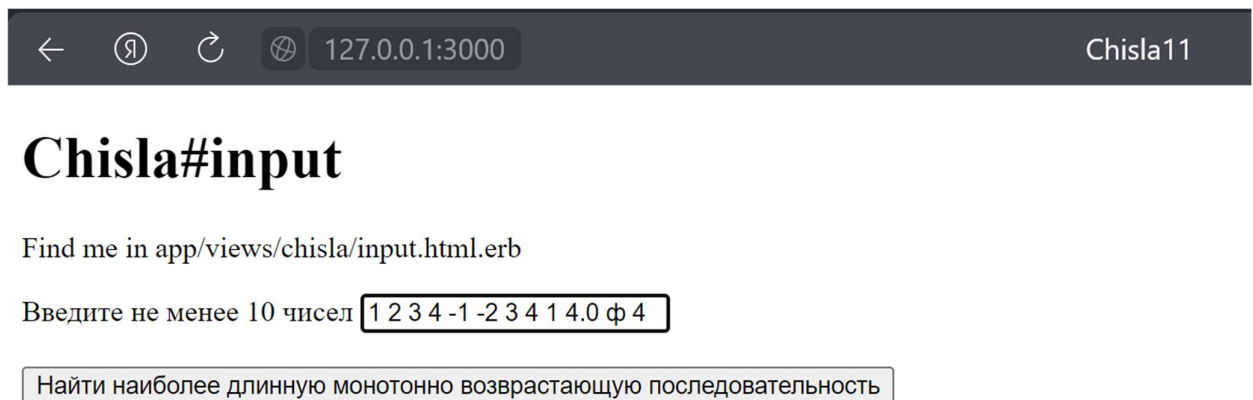


Рисунок 1 – вид страницы chisla_input

Chisla#view

Find me in app/views/chisla/view.html.erb

Таблица результатов

Изначальный	Все возможные	Самая длинная
1 2 3 4 -1 -2 3 4 1 4 4	1 2 3 4	+
	-1	
	-2 3 4	
	1 4	

[Найти для других чисел](#)

Рисунок 2 – вид страницы chisla_view

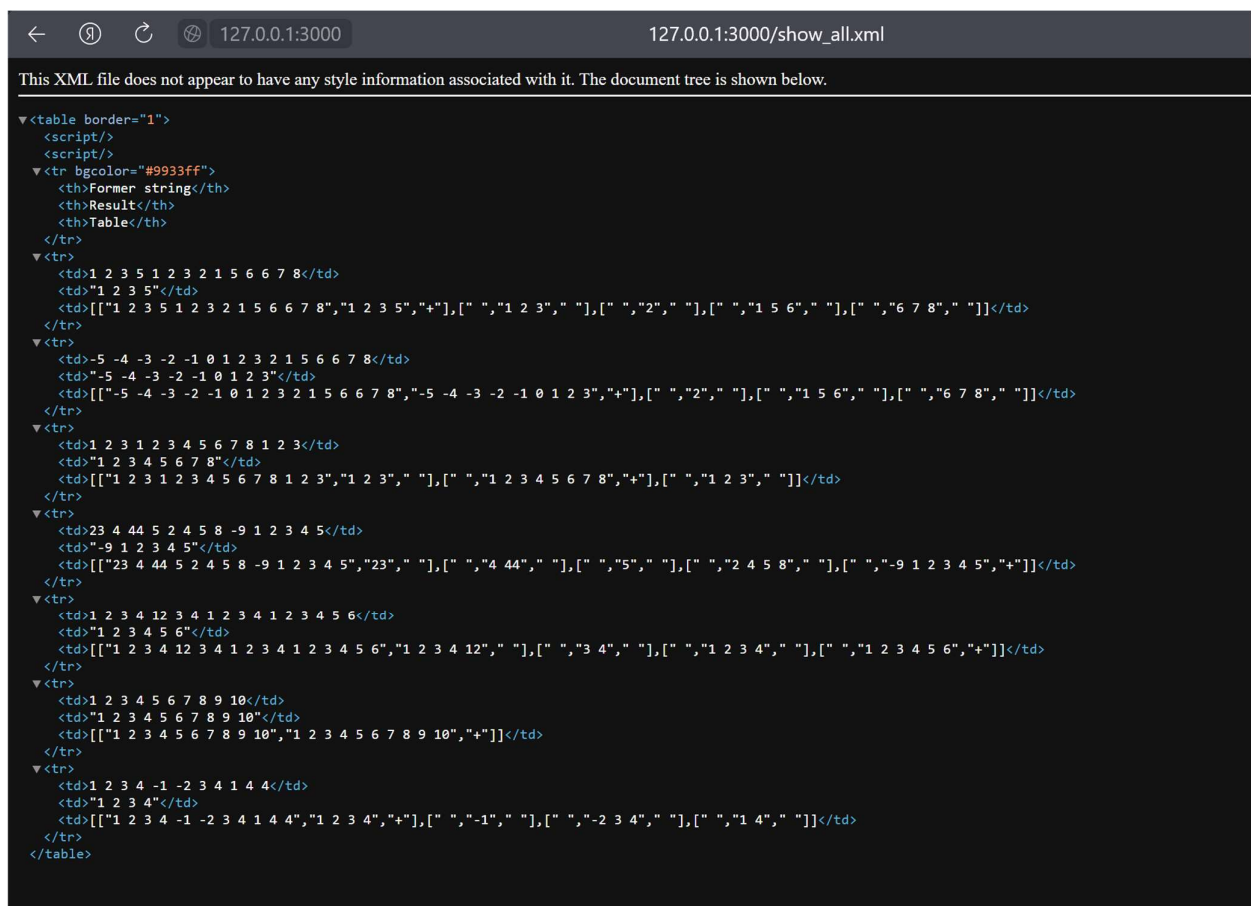


Рисунок 3 – вывод содержимого БД через сериализацию в XML

```
Parameters: {"str"=>"1 2 3 4 5 6 7 8 9 10"}
ChislaResult Load (0.2ms) SELECT "chisla_results".* FROM "chisla_results" WHERE "chisla_results"."string" = ? LIMIT ? [{"string", "1 2 3 4 5 6 7 8 9 10"}, [{"LIMIT", 1}]]
↳ app/controllers/chisla_controller.rb:22:in `check'
"Результат просчитан впервые"
TRANSACTION (0.1ms) begin transaction
↳ app/controllers/chisla_controller.rb:22:in `check'
ChislaResult Create (7.7ms) INSERT INTO "chisla_results" ("string", "result", "my_table", "created_at", "updated_at") VALUES (?, ?, ?, ?, ?)
[["string", "1 2 3 4 5 6 7 8 9 10"], ["result", "\1 2 3 4 5 6 7 8 9 10\\"], ["my_table", "\["1 2 3 4 5 6 7 8 9 10\\",\\"1 2 3 4 5 6 7 8 9 10\\",\\"+\\"]\\""], ["created_at", "2023-11-23 17:39:53.075576"], ["updated_at", "2023-11-23 17:39:53.075576"]]
↳ app/controllers/chisla_controller.rb:29:in `check'
TRANSACTION (1.9ms) commit transaction
↳ app/controllers/chisla_controller.rb:29:in `check'
Запись добавлена в БД #<ChislaResult:0x000001f2d8771ce0>
```

Рисунок 4 – вывод сообщения в консоль о добавлении ранее не существовавшей в БД записи

```
Parameters: {"str"=>"1 2 3 4 5 6 7 8 9 10"}
ChislaResult Load (0.2ms) SELECT "chisla_results".* FROM "chisla_results" WHERE "chisla_results"."string" = ? LIMIT ? [{"string", "1 2 3 4 5 6 7 8 9 10"}, [{"LIMIT", 1}]]
↳ app/controllers/chisla_controller.rb:22:in `check'
"Результат уже посчитан"
```

Рисунок 5 – вывод сообщения в консоли о поиске и нахождении предыдущего результата

```
Finished in 2.413526s, 2.9003 runs/s, 2.9003 assertions/s.
7 runs, 7 assertions, 0 failures, 0 errors, 0 skips
PS: P:\education\7_semester\lab1\lab11\lab11_y\chisla11>
```

Рисунок 7 – результат выполнения тестов

Распечатка БД:

```
<?xml version="1.0" encoding="System"?>
<table>
  <database></database>
  <name>chisla_results</name>
  <ddl><![CDATA[CREATE TABLE "chisla_results" ("id" integer PRIMARY
KEY AUTOINCREMENT NOT NULL, "string" text, "result" text, "my_table" json,
"created_at" datetime(6) NOT NULL, "updated_at" datetime(6) NOT NULL);]]></ddl>
  <columns>
    <column>
      <name>id</name>
      <type>integer</type>
      <constraints>
        <constraint>
          <type>PRIMARY KEY</type>
```

```

        <definition>PRIMARY KEY AUTOINCREMENT </definition>
    </constraint>
    <constraint>
        <type>NOT NULL</type>
        <definition>NOT NULL</definition>
    </constraint>
</constraints>
</column>
<column>
    <name>string</name>
    <type>text</type>
</column>
<column>
    <name>result</name>
    <type>text</type>
</column>
<column>
    <name>my_table</name>
    <type>json</type>
</column>
<column>
    <name>created_at</name>
    <type>datetime</type>
    <constraints>
        <constraint>
            <type>NOT NULL</type>
            <definition>NOT NULL</definition>
        </constraint>
    </constraints>
</column>

```

```

<column>
  <name>updated_at</name>
  <type>datetime</type>
  <constraints>
    <constraint>
      <type>NOT NULL</type>
      <definition>NOT NULL</definition>
    </constraint>
  </constraints>
</column>
</columns>
<rows>
  <row>
    <value column="0">3</value>
    <value column="1">1 2 3 5 1 2 3 2 1 5 6 6 7 8</value>
    <value column="2">&quot;1 2 3 5&quot;</value>
    <value column="3"><![CDATA["["["1 2 3 5 1 2 3 2 1 5 6 6 7 8\", \"1 2 3
5\", \"+\", [\" \", \"1 2 3\", \" \"], [\" \", \"2\", \" \"], [\" \", \"1 5 6\", \" \"], [\" \", \"6 7 8\", \"
\"]]]"]]]></value>
    <value column="4">2023-11-20 18:26:06.265756</value>
    <value column="5">2023-11-20 18:26:06.265756</value>
  </row>
  <row>
    <value column="0">4</value>
    <value column="1">-5 -4 -3 -2 -1 0 1 2 3 2 1 5 6 6 7 8</value>
    <value column="2">&quot;-5 -4 -3 -2 -1 0 1 2 3&quot;</value>
    <value column="3"><![CDATA["["["-5 -4 -3 -2 -1 0 1 2 3 2 1 5 6 6 7
8\", \"-5 -4 -3 -2 -1 0 1 2 3\", \"+\", [\" \", \"2\", \" \"], [\" \", \"1 5 6\", \" \"], [\" \", \"6 7 8\", \"
\"]]]"]]]></value>
    <value column="4">2023-11-20 18:26:14.793879</value>

```



```

        <value column="5">2023-11-20 18:26:14.793879</value>
    </row>
    <row>
        <value column="0">5</value>
        <value column="1">1 2 3 1 2 3 4 5 6 7 8 1 2 3</value>
        <value column="2">&quot;1 2 3 4 5 6 7 8&quot;</value>
        <value column="3"><![CDATA["["["1 2 3 1 2 3 4 5 6 7 8 1 2 3\", \"1 2
3\", \" \"],[\" \", \"1 2 3 4 5 6 7 8\", \"+\",[\" \", \"1 2 3\", \" \"]]"]]></value>
        <value column="4">2023-11-23 17:24:15.411602</value>
        <value column="5">2023-11-23 17:24:15.411602</value>
    </row>
    <row>
        <value column="0">6</value>
        <value column="1">23 4 44 5 2 4 5 8 -9 1 2 3 4 5</value>
        <value column="2">&quot;-9 1 2 3 4 5&quot;</value>
        <value column="3"><![CDATA["["["23 4 44 5 2 4 5 8 -9 1 2 3 4
5\", \"23\", \" \"],[\" \", \"4 44\", \" \"],[\" \", \"5\", \" \"],[\" \", \"2 4 5 8\", \" \"],[\" \", \"-9 1 2 3
4 5\", \"+\"]]"]]></value>
        <value column="4">2023-11-23 17:25:05.623126</value>
        <value column="5">2023-11-23 17:25:05.623126</value>
    </row>
    <row>
        <value column="0">7</value>
        <value column="1">1 2 3 4 12 3 4 1 2 3 4 1 2 3 4 5 6</value>
        <value column="2">&quot;1 2 3 4 5 6&quot;</value>
        <value column="3"><![CDATA["["["1 2 3 4 12 3 4 1 2 3 4 1 2 3 4 5 6\", \"1
2 3 4 12\", \" \"],[\" \", \"3 4\", \" \"],[\" \", \"1 2 3 4\", \" \"],[\" \", \"1 2 3 4 5
6\", \"+\"]]"]]></value>
        <value column="4">2023-11-23 17:39:06.222681</value>
        <value column="5">2023-11-23 17:39:06.222681</value>

```

```

</row>
<row>
  <value column="0">8</value>
  <value column="1">1 2 3 4 5 6 7 8 9 10</value>
  <value column="2">"1 2 3 4 5 6 7 8 9 10"</value>
  <value column="3">"["<"1 2 3 4 5 6 7 8 9 10\<">1
2 3 4 5 6 7 8 9 10\<">+<">]<"></value>
  <value column="4">2023-11-23 17:39:53.075576</value>
  <value column="5">2023-11-23 17:39:53.075576</value>
</row>
<row>
  <value column="0">9</value>
  <value column="1">1 2 3 4 -1 -2 3 4 1 4 4</value>
  <value column="2">"1 2 3 4"</value>
  <value column="3"><![CDATA["["<"1 2 3 4 -1 -2 3 4 1 4 4",<">1 2 3
4",<">+<">],<">","<-1",<">"],<">","<-2 3 4",<">"],<">","<1 4",<">"]<">]></value>
  <value column="4">2023-11-23 18:15:07.363712</value>
  <value column="5">2023-11-23 18:15:07.363712</value>
</row>
<row>
  <value column="0">10</value>
  <value column="1">-1 -2 -3 2 3 4 -2 -3 2 3</value>
  <value column="2">"-3 2 3 4"</value>
  <value column="3"><![CDATA["["<-1 -2 -3 2 3 4 -2 -3 2 3\<">","<-1",<">
\<">],<">","<-2\<">"],<">","<-3 2 3 4\<">+<">],<">","<-2\<">"],<">","<-3 2 3\<">,"
\<">"]<">]></value>
  <value column="4">2023-11-24 07:55:53.499932</value>
  <value column="5">2023-11-24 07:55:53.499932</value>
</row>
</rows>

```

</table>

Вывод: были изучены базы данных и работа с ними на языке Ruby, были сделаны проверка наличия уже имеющихся записей в ней, добавление новых и использование данные из неё.