Московский авиационный институт

(Национальный исследовательский университет)

Институт “Информационные технологии и прикладная математика”

**Курсовой проект**

по курсу «Базы данных»

Тема “Лабораторные медицинские исследования”

**Студенты:** Гусева С. Р., Николаева Е. С., Панкратова К. И.

**Группа:** М8О-301Б-20

**Преподаватель:** Лукин В.Н.

**Оценка:**

Москва, 2022

**Постановка задачи:**

Разработать пользовательское приложение, работающее на основе конкретной базы данных.

**1. Таблицы и их описание**

1. **Таблица patient**

CREATE TABLE patient(

id\_patient BIGINT NOT NULL PRIMARY KEY,

last\_name VARCHAR(50) NOT NULL,

first\_name VARCHAR(50) NOT NULL,

gender VARCHAR(6) NOT NULL,

date\_of\_birth DATE NOT NULL,

email VARCHAR(100));

INSERT INTO patient(

id\_patient, last\_name, first\_name, gender, date\_of\_birth, email);

В таблице пациентов (patient) у каждого пациента есть свой четырехзначный id (который является первичным ключом), фамилия, имя, пол, дата рождения и электронная почта (которая может быть не заполнена).

*Пример заполнения:*

INSERT INTO patient VALUES(1,'Flack','Roger','Male','5/21/1960','');

1. **Таблица laborant**

CREATE TABLE laborant(

id\_laborant BIGINT NOT NULL PRIMARY KEY,

last\_name VARCHAR(50) NOT NULL,

first\_name VARCHAR(50) NOT NULL,

gender VARCHAR(6) NOT NULL,

beggining\_of\_work DATE NOT NULL,

email VARCHAR(100));

INSERT INTO laborant(

id\_laborant, last\_name, first\_name, beggining\_of\_work, email);

В таблице лаборантов (laborant) у каждого лаборанта есть свой трехзначный id (который является первичным ключом), фамилия, имя, пол, дата начала работы и электронная почта (которая может быть не заполнена).

*Пример заполнения:*

INSERT INTO laborant VALUES(1,'Price','Barney','Male','2/26/2011','[Barney\_Price3210@bauros.biz](mailto:Barney_Price3210@bauros.biz)');

1. **Таблица branch**

CREATE TABLE branch(

number\_branch BIGINT NOT NULL PRIMARY KEY,

adress VARCHAR(50) NOT NULL,

working\_hours VARCHAR(10) NOT NULL,

telephone\_number VARCHAR(15) NOT NULL);

INSERT INTO branch(

number\_branch, adress, working\_hours, telephone\_number);

В таблице филиалов (branch) у каждого филиала есть свой номер (что является первичным ключом), адрес, часы работы и номер телефона.

*Заполнение таблицы:*

VALUES('1', 'Moscow, Vorontsovskaya street, 19A, building 1', '08:00-21:00', '8-495-780-07-71')

VALUES('2', 'Moscow, General Karbyshev Boulevard, 13, building 1', '08:00-20:00', '8-495-780-07-72')

VALUES('3', 'Moscow, Barrikadnaya street, 19, building 3', '09:00-21:00', '8-495-780-07-73')

VALUES('4', 'Moscow, Marshal Tukhachevsky, 37/21', '09:00-21:00', '8-495-780-07-72')

VALUES('5', 'Moscow, Sevastopolsky prospekt, 10, building 2', '08:00-21:00', '8-495-780-07-74')

VALUES('6', 'Moscow, Michurinsky Prospekt, 34', '09:00-21:00', '8-495-780-07-79')

VALUES('7', 'Moscow, Proezd Dezhneva, 1', '08:00-21:00', '8-495-780-07-75')

VALUES('8', 'Moscow, Nizhnyaya Pervomayskaya street, 43A', '08:00-21:00', '8-495-780-07-78')

1. **Таблица analysis**

CREATE TABLE analysis(

number\_analysis BIGINT NOT NULL PRIMARY KEY,

type VARCHAR(50) NOT NULL,

price FLOAT NOT NULL,

norm VARCHAR(20) NOT NULL

);

В таблице анализов (analysis) у каждого анализа есть свой порядковый номер, тип анализа (все типы представлены в таблице ниже), цена и показатель нормы)

|  |  |  |
| --- | --- | --- |
| Type | Price | Norm |
| Leukocytes | 100.00 | 4,5-11 |
| Lymphocytes | 150.00 | 19-37 |
| Erythrocytes | 90.00 | 3,8-5,1 |
| Hemoglobin | 120.00 | 117-155 |
| Platelets | 145.00 | 150-450 |
| Monocytes | 135.00 | 3-11 |

*Заполнение таблицы:*

INSERT INTO analysis VALUES(1, 'Leukocytes', '100.00', '4,5-11');

INSERT INTO analysis VALUES(2, 'Lymphocytes', '150.00', '19-37');

INSERT INTO analysis VALUES(3, 'Erythrocytes', '90.00', '3,8-5,1');

INSERT INTO analysis VALUES(4, 'Hemoglobin', '120.00', '117-155');

INSERT INTO analysis VALUES(5, 'Platelets', '145.00', '150-450');

INSERT INTO analysis VALUES(6, 'Monocytes', '135.00', '3-11');

1. **Таблица research**

CREATE TABLE research(

number\_research BIGINT NOT NULL PRIMARY KEY,

id\_laborant BIGINT NOT NULL,

number\_analysis BIGINT NOT NULL,

number\_branch BIGINT NOT NULL

);

В таблице исследований есть номер результата(первичный ключ), id лаборанта, номер проводимого анализа (так как у всех анализов есть закрепленные за ними номера) и номер филиала, в котором будут браться эти анализы.

*Пример заполнения:*

INSERT INTO research VALUES('1', '1', '1', '8');

INSERT INTO research VALUES('2', '3', '1', '3');

1. **Таблица result**

CREATE TABLE result (

number\_result BIGINT NOT NULL PRIMARY KEY,

id\_patient BIGINT NOT NULL,

number\_research BIGINT NOT NULL,

date DATE NOT NULL,

result VARCHAR(50),

conclusion VARCHAR(50) NOT NULL

);

ALTER TABLE result

ADD CONSTRAINT fk\_patient

FOREIGN KEY (id\_patient)

REFERENCES patient(id\_patient);

ALTER TABLE result

ADD CONSTRAINT fk\_nresearch

FOREIGN KEY (number\_research)

REFERENCES research(number\_research);

У каждого результата есть номер результата(первичный ключ), id пациента, номер исследования, дата проведения исследования, результат и заключение (в данном случае, входит данный результат в номер проводимого анализа или нет). Расшифровки заключений представлены ниже:

within the normal range - входит в рамки нормы

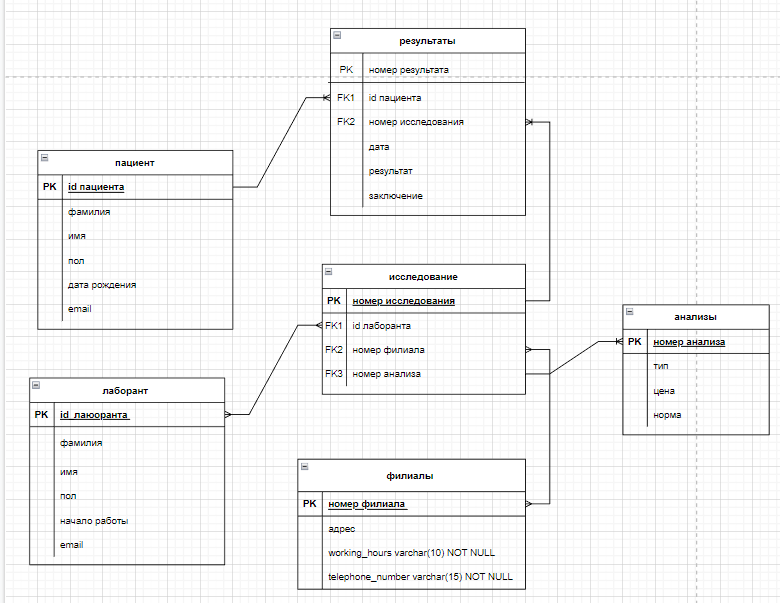
above normal - выше нормы

below normal - ниже нормы

*Пример заполнения:*

INSERT INTO result VALUES('1', '3', '5', '10/23/2022', '3,6', 'below normal');

**Логическая модель:**

****

**2. Код программы:**

library(DBI)

library(RPostgres)

library(DT)

library(shiny)

con <- dbConnect(RPostgres::Postgres(),

dbname = "datab",

host = "localhost",

user = "postgres",

password = "1234",

port = "5432"

)

maxmin\_idp <- c(dbGetQuery(con,

"SELECT MAX(id\_patient), MIN(id\_patient) from patient;"))

maxmin\_yb <- c(dbGetQuery(con,

"SELECT MAX(year\_of\_birth), MIN(year\_of\_birth) from patient;"))

maxmin\_idl <- c(dbGetQuery(con,

"SELECT MAX(id\_laborant), MIN(id\_laborant) from laborant;"))

maxmin\_rslt <- c(dbGetQuery(con,

"SELECT MAX(number\_result), MIN(number\_result) from result;"))

maxmin\_srch <- c(dbGetQuery(con,

"SELECT MAX(number\_research), MIN(number\_research) from research;"))

maxmin\_br <- c(dbGetQuery(con,

"SELECT MAX(number\_branch), MIN(number\_branch) from branch;"))

maxmin\_nan <- c(dbGetQuery(con,

"SELECT MAX(number\_analysis), MIN(number\_analysis) from analysis;"))

ui <- navbarPage(

theme = bslib::bs\_theme(bootswatch = "lux"),

"База данных \"Лабораторные медицинские исследования\"",

tabPanel("Результаты",

sidebarLayout(

sidebarPanel( textInput("rslt1", "Номер результата"),

textInput("patient", "ID пациента"),

textInput("rslt3", "Номер исследования"),

textInput("rslt4", "Дата"),

textInput("rslt5", "Результат исследования"),

textInput("rslt6", "Заключение"),

),

mainPanel(

DT::dataTableOutput("table1")

)

)

),

tabPanel("Пациенты",

sidebarLayout(

sidebarPanel(

sliderInput("year\_selector","Год рождения пациента",

min = maxmin\_yb$min,

max = maxmin\_yb$max,

value = c(maxmin\_yb$min, maxmin\_yb$max),

step = 1

),

textInput("pat1", "ID пациента"),

textInput("pat2", "Фамилия"),

textInput("pat3", "Имя"),

selectInput("checkGroup\_g", label = "Пол",

choices = list("Все" = 1, "Male", "Female"),

selected = 1),

textInput("pat5", "Год рождения"),

textInput("pat6", "Email"),

actionButton("do\_pat", "Добавить запись")

),

mainPanel(

DT::dataTableOutput("table2")

)

)

),

tabPanel("Исследования",

sidebarLayout(

sidebarPanel(

textInput("srch1", "Номер исследования"),

textInput("srch2", "ID пациента"),

selectInput("checkGroup\_an", label = "Номер анализа",

choices = list("Все анализы" = 1, "1", "2", "3", "4", "5", "6"),

selected = 1),

textInput("srch4", "Номер филиала"),

actionButton("do\_srch", "Добавить запись")

),

mainPanel(

DT::dataTableOutput("table3")

)

)

),

tabPanel("Лаборанты",

sidebarLayout(

sidebarPanel(

textInput("lab1", "ID лаборанта"),

textInput("lab2", "Фамилия"),

textInput("lab3", "Имя"),

selectInput("checkGroup\_g", label = "Пол",

choices = list("Все" = 1, "Мужской", "Женский"),

selected = 1),

textInput("lab5", "Дата начала работы"),

textInput("lab6", "Email"),

actionButton("do\_lab", "Добавить запись")

),

mainPanel(

DT::dataTableOutput("table4")

)

)

),

)

server <- function(input, output, session) {

output$table1 <- DT::renderDataTable({

sql\_1 <- "SELECT \* from result"

query <- sqlInterpolate(ANSI(), sql\_1)

outp <- dbGetQuery(con, query)

ret <- DT::datatable(outp)

return(ret)

})

output$table2 <- DT::renderDataTable({

# sql\_2 <- "SELECT \* from patient"

# query <- sqlInterpolate(ANSI(), sql\_2)

sql\_4 <- "SELECT \* from patient

WHERE year\_of\_birth

BETWEEN ?yb\_min and ?yb\_max;"

query <- sqlInterpolate(ANSI(), sql\_4,

yb\_min = input$year\_selector[1],

yb\_max = input$year\_selector[2])

outp <- dbGetQuery(con, query)

ret <- DT::datatable(outp)

return(ret)

})

output$table3 <- DT::renderDataTable({

sql\_3 <- "SELECT \* from research"

query <- sqlInterpolate(ANSI(), sql\_3)

outp <- dbGetQuery(con, query)

ret <- DT::datatable(outp)

return(ret)

})

output$table4 <- DT::renderDataTable({

sql\_4 <- "SELECT \* from laborant"

query <- sqlInterpolate(ANSI(), sql\_4)

outp <- dbGetQuery(con, query)

ret <- DT::datatable(outp)

return(ret)

})

observeEvent(input$do\_pat, {

dbExecute(con, "INSERT INTO patient VALUES ($1, $2, $3, $4, $5, $6)",

params = list(input$pat1, input$pat2, input$pat3, input$checkGroup\_g, input$pat5, input$pat6))

})

observeEvent(input$do\_lab, {

dbExecute(con, "INSERT INTO laborant VALUES ($1, $2, $3, $4, $5, $6)",

params = list(input$laborant, input$lab2, input$lab3, input$checkGroup\_g, input$lab5, input$lab6))

})

observeEvent(input$do\_rslt, {

dbExecute(con, "INSERT INTO result VALUES ($1, $2, $3, $4, $5, $6)",

params = list(input$rslt1, input$rslt2, input$rslt3, input$rslt4, input$rslt5, input$rslt6))

})

observeEvent(input$do\_srch, {

dbExecute(con, "INSERT INTO research VALUES ($1, $2, $3, $4)",

params = list(input$srch1, input$srch2, input$srch3, input$srch4))

})

}

shinyApp(ui = ui, server = server)

**3. Описание функций, использованных в программе:**

**textInput()** – функция ввода небольшого фрагмента текста

Данная функция считывает информацию, записанную пользователем в строке “номер рузельтата” и запоминает ее в переменной rslt1:

textInput("rslt1", "Номер результата)

**sliderInput()** – функция, которая отвечает за ползунок

В данном случае функция отображает ползунок, которые можете перемещаться от наименьшего значения id пациента до наибольшего с шагом 1:

sliderInput("patient", label = "ID пациента",

min = maxmin\_idp$min,

max = maxmin\_idp$max,

value = c(maxmin\_idp$min, maxmin\_idp$max),

step = 1)

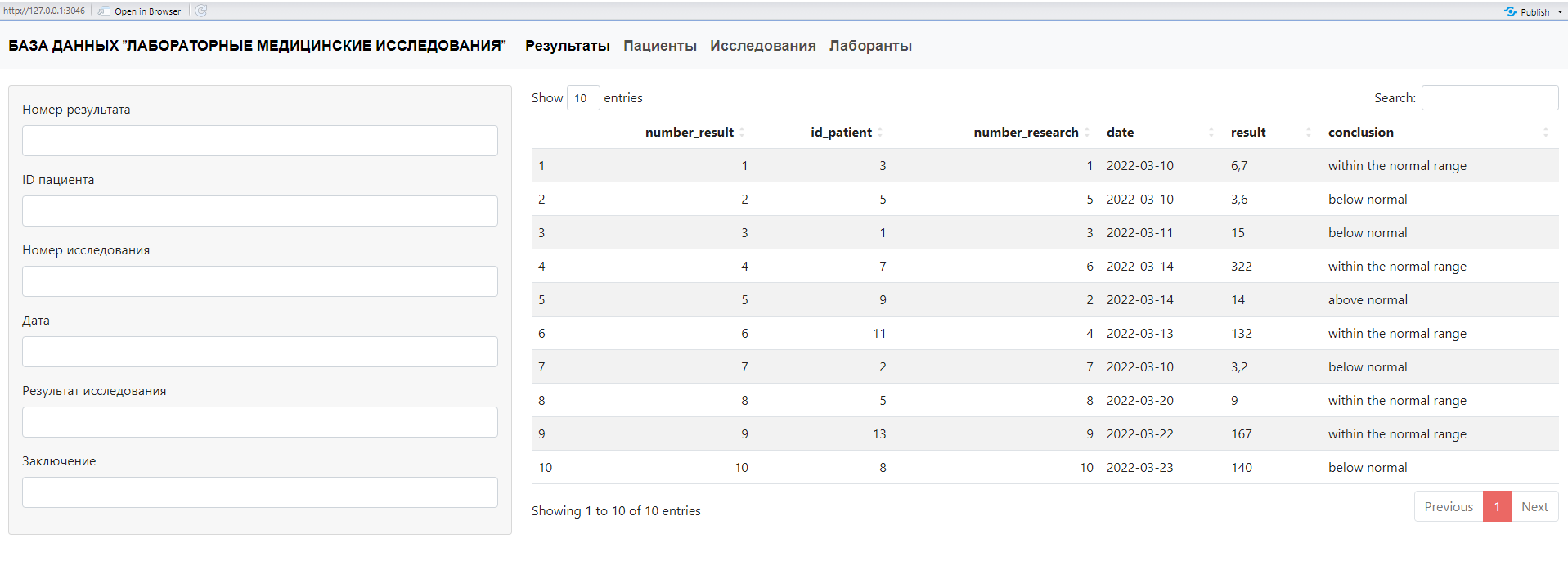
**actionButton()** – функция, подтверждающая действия пользователя, которая создает кнопку для нажатия.

Следующая функция создает кнопку “Создать запись”, на которую пользователь может нажать и сохранить тем самым все введенные им данные про результат исследования в соответствующую таблицу:

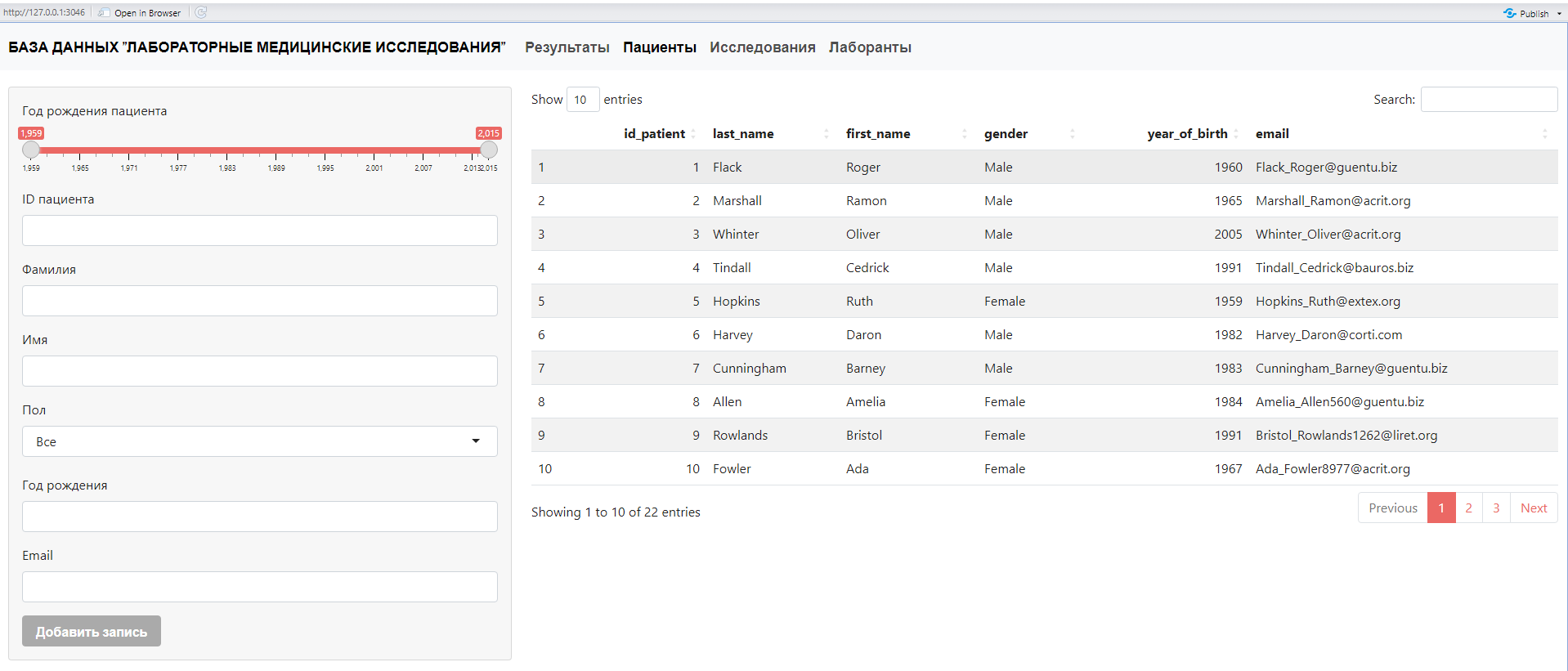
actionButton("do\_rslt", "Добавить запись о проведенном исследовании")

**4. Результат работы программы:**

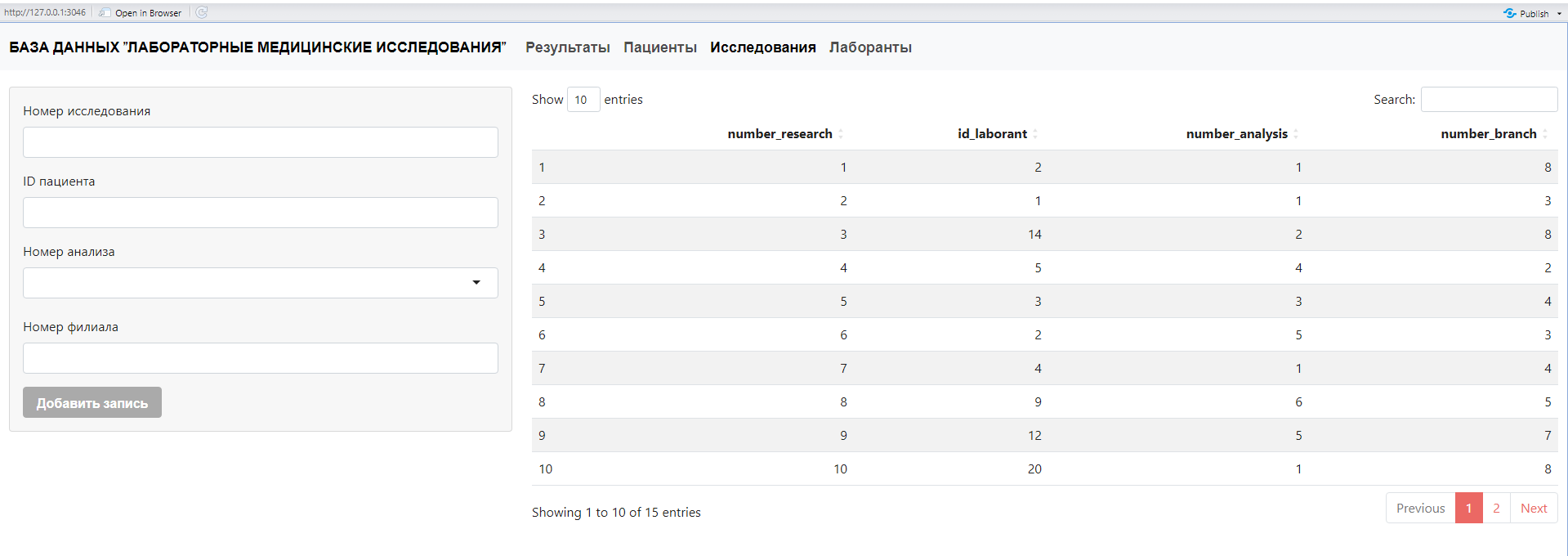
**Таблица результатов**

****

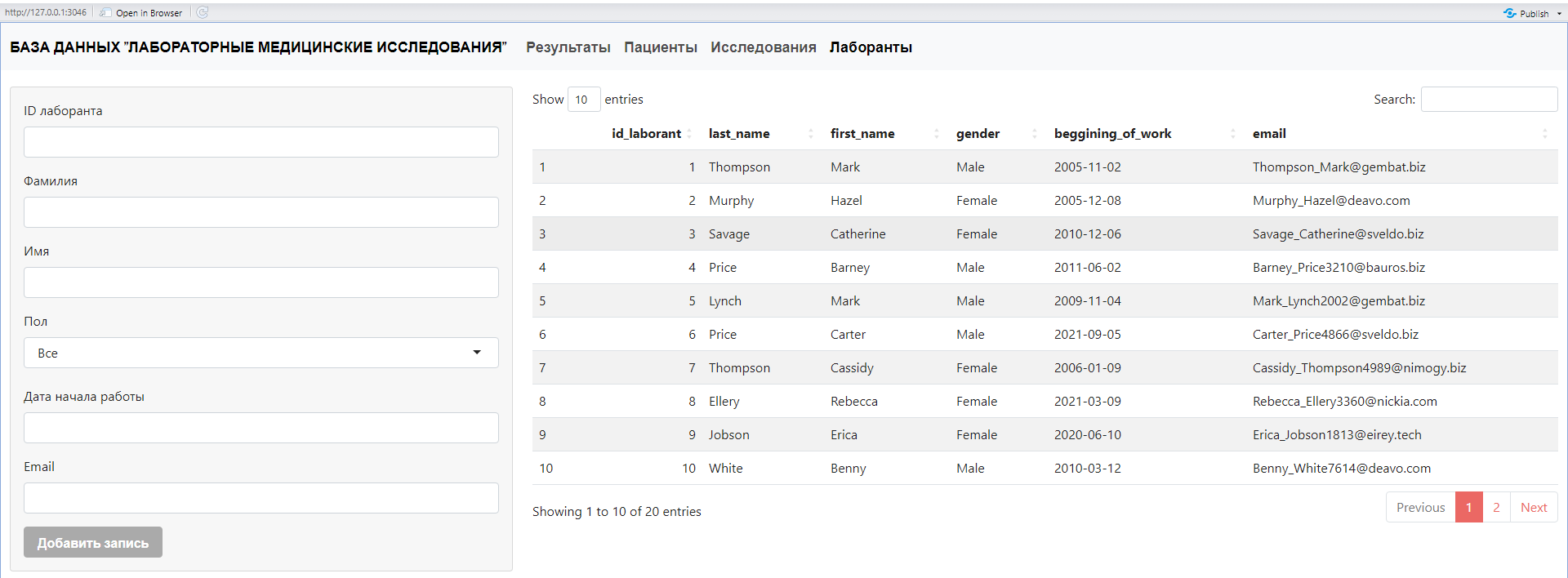
**Таблица пациентов**

****

**Таблица Исследований**

****

**Таблица лаборантов**

****

**Список литературы:**

1. Хэндли Уикем. “Изучаем Shiny. Строим интерактивные приложения, отчеты и дашборды на языке R”. (последняя даты обращения: 16.12.2022)
2. В. Н. Лукин. “Введение в проектирование баз данных”. (последняя дата обращения: 16.12.2022)

**Итоговые оценки:**

Общее количество баллов: 13

Панкратова Ксения: 5

Гусева Софья: 5

Николаева Елизавета: 3