МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования  
**«Вятский государственный университет»**

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

**«Разработка приложения на ЯП С# с базой данных под управлением PostgreSQL»**

Отчёт по лабораторной работе №5 дисциплины

«Базы данных»

Выполнил студент группы ИВТб-31\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Жеребцов К.А./

Проверил преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Клюкин В. Л./

Киров 2023

**Цели лабораторной работы:**

Познакомиться c библиотекой в C# для связывания приложения с БД;

Изучить некоторые шаблоны проектирования, связанные с работой с БД;

Освоить на практике основы взаимодействия с БД под управлением PostgreSQL в приложении на C#.

**Задание на лабораторную работу:**

1. При выполнении работы нужно использовать БД, созданную в предыдущих лабораторных работах. Необходимо создать приложение на языке программирования С# с графическим интерфейсом. Для любой одной таблицы в БД, содержащей хотя бы один внешний ключ на другую таблицу, приложение должно:
2. Выводить строки выбранной таблицы;
3. Предоставлять любой фильтр по значениям строк (например, «Дата с … по …»);
4. Предоставлять возможность добавить новую строку в таблицу;
5. Предоставлять возможность удалить строку из таблицы. Требования к реализации:
6. В выводе таблицы должны показываться осмысленные значения (например, если таблица «Чек» ссылается на таблицу «Товар», нужно вывести не id товара, а его название);
7. При добавлении новой записи пользователь также должен не ввести значение внешнего ключа, а выбрать строку из списка;
8. Или сохранение, или удаление строки должно быть реализовано с помощью функции на PL/pgSQL;
9. Фильтроваться значения должны с помощью SQL-запроса, а не в готовой коллекции на клиенте; При разработке нужно использовать шаблоны проектирования, связанные с работой с БД. Отчет должен содержать диаграмму разработанных классов, организующих взаимодействие с БД, их исходный код, а также экранные формы работы приложения.

**Ход работы:**

1. Функция вывода таблицы public.teacher

string sql = "Server = 127.0.0.1; Port = 5432; User Id = postgres; Password = 1234; Database = laba";

// Установка соединения с БД

NpgsqlConnection conn = new NpgsqlConnection(sql);

NpgsqlCommand com = new NpgsqlCommand("SELECT t.\* FROM public.teacher t",conn);

conn.Open();

NpgsqlDataReader reader = com.ExecuteReader();

while (reader.Read())

{

try

{

dataGridView1.Rows.Add(reader.GetInt32(0).ToString(), reader.GetString(1), reader.GetString(2), reader.GetString(3), reader.GetString(4));

}

catch {}

}

// Тут можно делать запросы, используя соединение

// Всегда соединение должно быть в итоге закрыто!

conn.Close();

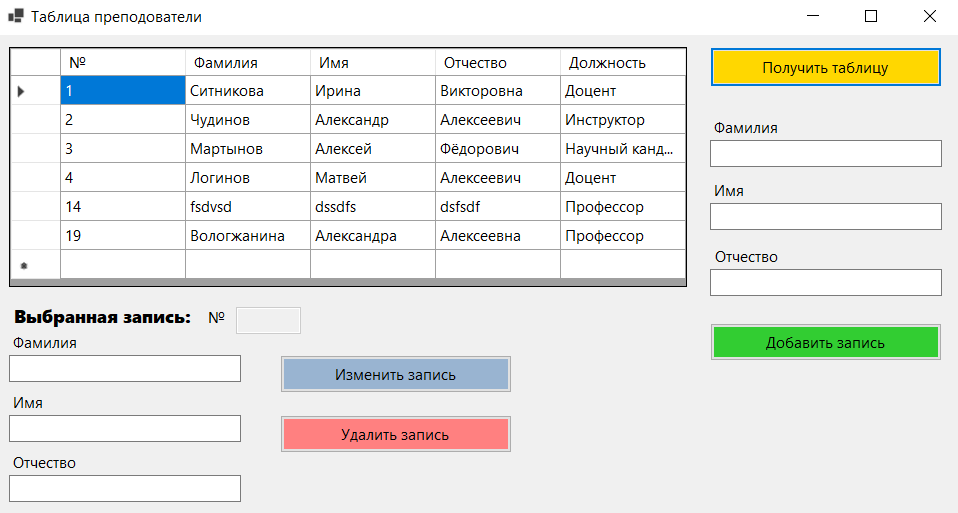
****

Рисунок 1 – Выведенные данные из таблицы teacher

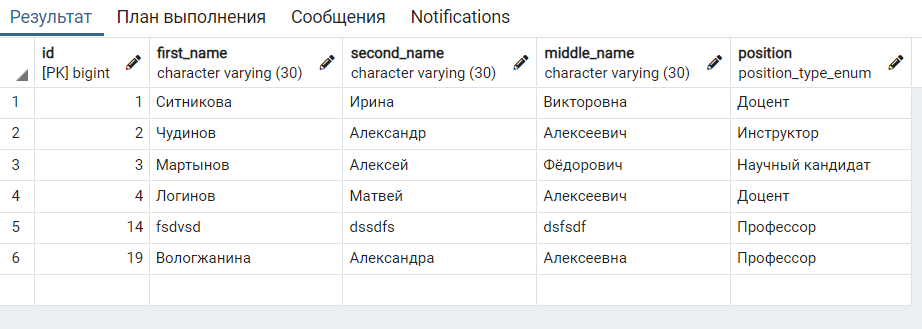


Рисунок 2 – Данные в таблице teacher в PostgreSQL

1. Функция добавления записи таблицы public.teacher

string sql = "Server = 127.0.0.1; Port = 5432; User Id = postgres; Password = 1234; Database = laba";

// Установка соединения с БД

NpgsqlConnection con = new NpgsqlConnection(sql);

con.Open();

NpgsqlCommand com = new NpgsqlCommand("INSERT INTO teacher(first\_name, second\_name, middle\_name, position) VALUES(@p1, @p2, @p3, 'Доцент')", con);

var a = new NpgsqlParameter("@p1", NpgsqlTypes.NpgsqlDbType.Varchar);

var b = new NpgsqlParameter("@p2", NpgsqlTypes.NpgsqlDbType.Varchar);

var c = new NpgsqlParameter("@p3", NpgsqlTypes.NpgsqlDbType.Varchar);

a.Value = textBox1.Text;

b.Value = textBox2.Text;

c.Value = textBox3.Text;

com.Parameters.Add(a);

com.Parameters.Add(b);

com.Parameters.Add(c);

com.ExecuteNonQuery();

dataGridView1.Rows.Clear();

button1\_Click(this, e);

con.Close();

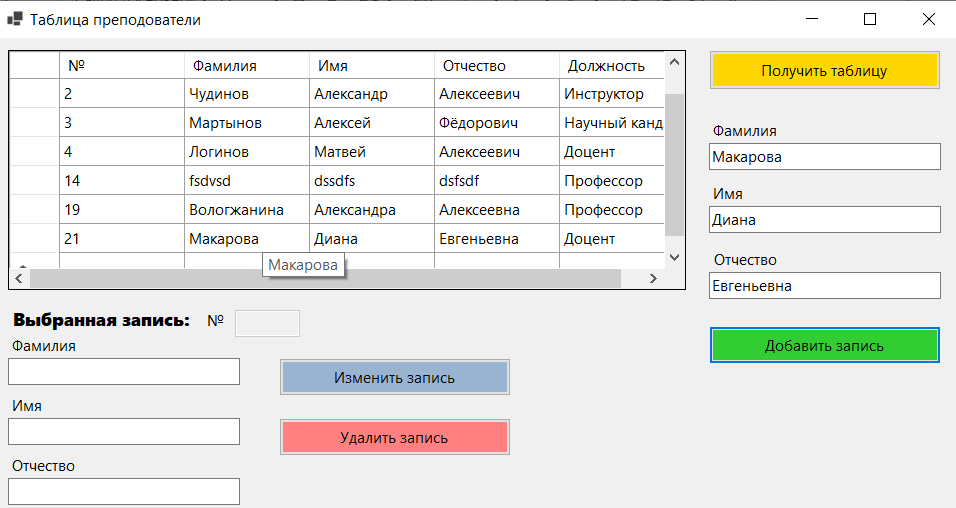
****

Рисунок 3 – Добавленная запись в таблицу teacher

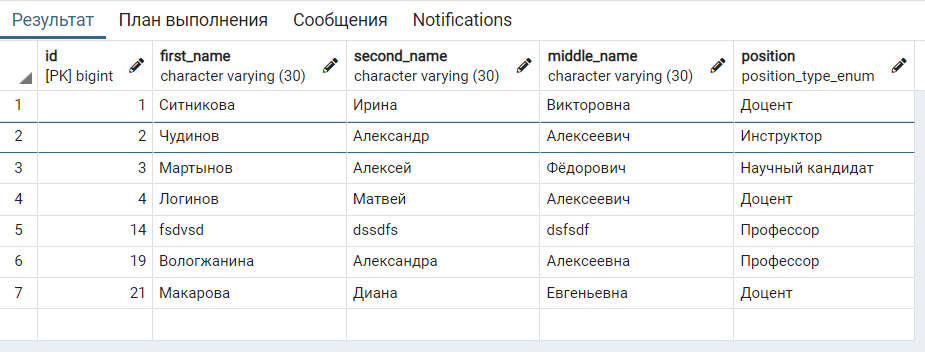


Рисунок 4 – Добавленная запись в таблицу teacher в PostgreSQL

1. Функция изменения записи таблицы public.teacher

string sql = "Server = 127.0.0.1; Port = 5432; User Id = postgres; Password = 1234; Database = laba";

NpgsqlConnection con = new NpgsqlConnection(sql);

con.Open();

NpgsqlCommand com = new NpgsqlCommand("UPDATE teacher SET first\_name = @p2,second\_name = @p3,middle\_name = @p4,position = 'Доцент' WHERE id = @p1", con);

var a = new NpgsqlParameter("@p1", NpgsqlTypes.NpgsqlDbType.Integer);

var b = new NpgsqlParameter("@p2", NpgsqlTypes.NpgsqlDbType.Varchar);

var c = new NpgsqlParameter("@p3", NpgsqlTypes.NpgsqlDbType.Varchar);

var d = new NpgsqlParameter("@p4", NpgsqlTypes.NpgsqlDbType.Varchar);

a.Value = Convert.ToInt32(textBox7.Text);

b.Value = textBox6.Text;

c.Value = textBox5.Text;

d.Value = textBox5.Text;

com.Parameters.Add(a);

com.Parameters.Add(b);

com.Parameters.Add(c);

com.Parameters.Add(d);

com.ExecuteNonQuery();

dataGridView1.Rows.Clear();

button1\_Click(this, e);

con.Close();

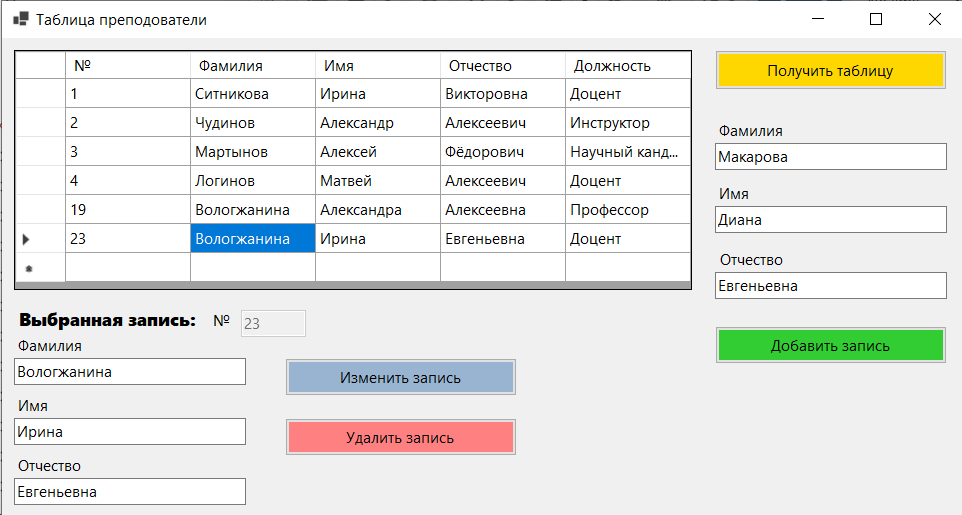
****

Рисунок 5 – Изменённая запись

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 – Изменённая запись в таблице teacher в PostgreSQL

1. Функция удаления записи таблицы public.teacher

string sql = "Server = 127.0.0.1; Port = 5432; User Id = postgres; Password = 1234; Database = laba";

try

{

NpgsqlConnection con = new NpgsqlConnection(sql);

con.Open();

NpgsqlCommand com = new NpgsqlCommand("DELETE FROM teacher WHERE id = @p1", con);

var a = new NpgsqlParameter("@p1", NpgsqlTypes.NpgsqlDbType.Integer);

int i = dataGridView1.SelectedCells[0].RowIndex;

a.Value = Convert.ToInt32(dataGridView1.Rows[i].Cells[0].Value);

int b = dataGridView1.CurrentRow.Index;

com.Parameters.Add(a);

com.ExecuteNonQuery();

dataGridView1.Rows.Clear();

button1\_Click(this, e);

con.Close();

}

catch

{

MessageBox.Show("Ячейку нельзя удалить, так как она задействована");

}

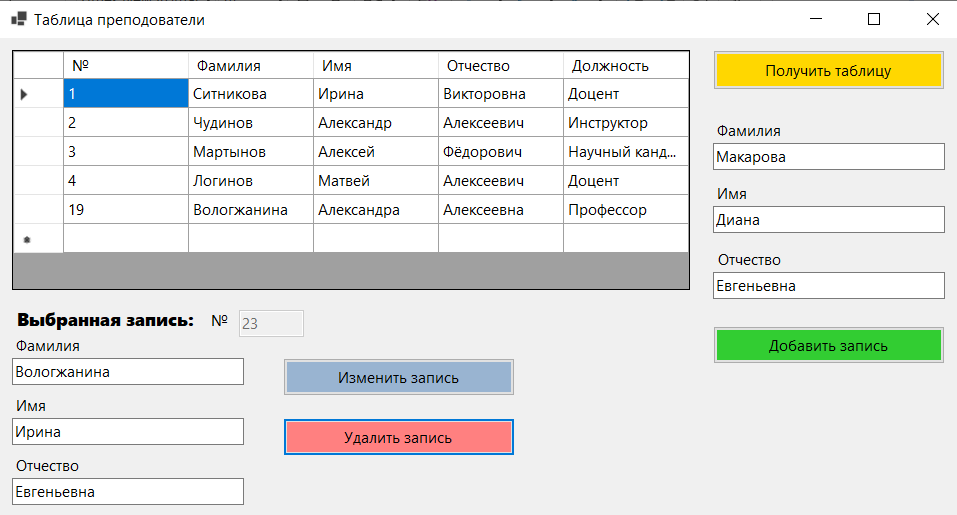


Рисунок 7 – Удалённая запись в таблице teacher

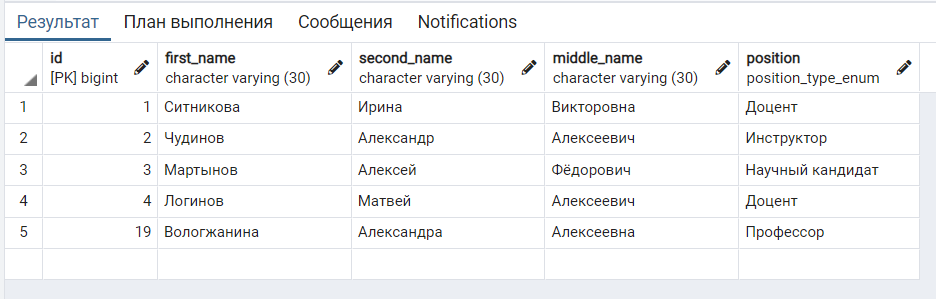


Рисунок 8 – Удалённая запись в таблице teacher в PostgreSQL

**Вывод:**

В ходе выполнения данной лабораторной работы познакомились c библиотекой в C# для связывания приложения с БД. Изучили некоторые шаблоны проектирования, связанные с работой с БД. Освоили на практике основы взаимодействия с БД под управлением PostgreSQL в приложении на C#.