МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики  
Кафедра информационных технологий

Отчёт по индивидуальному заданию

**по дисциплине «Разработка приложений в интегрированных средах»**

Вариант № 9

Работу выполнила  
студентка 35/2 группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. Е. Мостовая

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем

Проверил  
доцент кафедры, канд. техн. наук, доцент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. А. Полупанов

Краснодар 2024

**Цель работы:** разработать клиентское приложение на языке С# для работы с базой данных формата PostgreSQL. Приложение должно иметь возможность добавлять, изменять, удалять данные из БД. По каждому отчёту необходимо отобразить диаграммы подходящего типа, согласно варианту задания.

**Задание**. Предприятие реализует товары клиентам на условиях предоплаты. В договорах указываются товары, их количество и сумма. Оплата товаров может осуществляться как в наличном виде, путём внесения в кассу, так и перечислением. За период для выбранных клиентов сформировать материальный отчёт по товарам положенных к отгрузке, но не отгруженных с указанием количества и сумм (экспорт в MS Excel). За период для выбранных товаров сформировать отчёт, отображающий соотношение количества законтрактованных и отгруженных товаров.

**Практическая часть.**

На рисунке 1 представлено главное окно разработанной программы.

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Автоматически созданное описание

Рисунок 1 − Главное окно программы

На главном окне несколько вкладок для отображения данных из таблиц с информацией о клиентах, товарах и договорах с кнопками добавления изменения и удаления записей из таблиц, а также для создания диаграммы и отчёта в MS Excel.

Код подключения к базе данных:

private NpgsqlConnection con;

private string conString =

"Host = 127.0.0.1; Username = postgres; Password = 123; Database = company\_project";

Код создания таблицы с данными клиентов:

CREATE TABLE IF NOT EXISTS clients(

id SERIAL PRIMARY KEY NOT NULL,

lastname VARCHAR(20) NOT NULL,

name VARCHAR(20) NOT NULL,

surname VARCHAR(20) NOT NULL,

passport VARCHAR(10) NOT NULL,

phone VARCHAR(20) NOT NULL,

address VARCHAR(50) NOT NULL

);

Пример заполнения таблицы с данными клиентов:

INSERT INTO clients (lastname, name, surname,passport, phone, address) VALUES ("Мостовая", "Елизавета", "Евгеньевна", "0323111111", "89181111111", "ул. Железнодорожная");

Код для выгрузки данных в формате отчёта в MS Excel:

private void button1\_Click(object sender, EventArgs e)

{

string sql1 = "SELECT id, name FROM products";

NpgsqlCommand cmd1 = new NpgsqlCommand(sql1, con);

Excel.Application ex = new Excel.Application();

Workbook wb = ex.Workbooks.Open("C:/Users/Елизавета/Documents/Универ разное/РПвИС/list1.xlsx");

Worksheet sheet = wb.Sheets[1];

sheet.Cells[1, 1].Value = "Номер договора";

sheet.Cells[1, 2].Value = "Наименование товара";

sheet.Cells[1, 3].Value = "Количество товара";

sheet.Cells[1, 4].Value = "Сумма";

string ProductName = "";

DateTime d1 = dtpStart.Value;

DateTime d3 = dtpEnd.Value;

int countRows = 2;

for (int i = 0; i < dgvContracts.Rows.Count; i++)

if (Convert.ToBoolean(dgvContracts.Rows[i].Cells[7].Value) == false && Convert.ToDateTime(dgvContracts.Rows[i].Cells[3].Value) >= d1 && Convert.ToDateTime(dgvContracts.Rows[i].Cells[3].Value) <= d3)

{

NpgsqlDataReader reader1 = cmd1.ExecuteReader();

while (reader1.Read())

{

if (dgvContracts.Rows[i].Cells[2].Value.ToString() == reader1.GetValue(0).ToString())

{

ProductName = reader1.GetString(1);

}

}

reader1.Close();

sheet.Cells[countRows, 1].Value = dgvContracts.Rows[i].Cells[0].Value;

sheet.Cells[countRows, 2].Value = ProductName;

sheet.Cells[countRows, 3].Value = dgvContracts.Rows[i].Cells[4].Value;

sheet.Cells[countRows, 4].Value = dgvContracts.Rows[i].Cells[5].Value;

countRows++;

}

sheet.Rows[1].Style.HorizontalAlignment = HorizontalAlignment.Center;

sheet.Rows[1].Style.VerticalAlignment = VerticalAlignment.Center;

sheet.Cells.WrapText = true;

ex.Visible = true;

}

Код обработчика кнопки для добавления записей в таблицу с информацией о клиентах:

private void btnAdd\_Click(object sender, EventArgs e)

{

string sql1 = "INSERT INTO clients(lastname, name, surname, passport, phone, address) VALUES(@lastname, @name, @surname, @passport, @phone, @address)";

NpgsqlCommand cmd1 = new NpgsqlCommand(sql1, con);

cmd1.Parameters.AddWithValue("lastname", this.tbLastname.Text);

cmd1.Parameters.AddWithValue("name", this.tbName.Text);

cmd1.Parameters.AddWithValue("surname", this.tbSurname.Text);

cmd1.Parameters.AddWithValue("passport", this.tbPassport.Text);

cmd1.Parameters.AddWithValue("phone", this.tbPhone.Text);

cmd1.Parameters.AddWithValue("address", this.tbAddress.Text);

cmd1.Prepare();

cmd1.ExecuteNonQuery();

loadClients();

this.tbLastname.Text = "";

this.tbName.Text = "";

this.tbSurname.Text = "";

this.tbPassport.Text = "";

this.tbPhone.Text = "";

this.tbAddress.Text = "";

}

Код обработчика кнопки для изменения записей в таблице с информацией о клиентах:

private void btnChange\_Click(object sender, EventArgs e)

{

string sql = "UPDATE clients SET lastname=@lastname, name=@name, surname=@surname, passport=@passport, phone = @phone, address = @address WHERE id = @id";

NpgsqlCommand cmd = new NpgsqlCommand(sql, con);

cmd.Parameters.AddWithValue("lastname", this.tbLastname.Text);

cmd.Parameters.AddWithValue("name", this.tbName.Text);

cmd.Parameters.AddWithValue("surname", this.tbSurname.Text);

cmd.Parameters.AddWithValue("passport", this.tbPassport.Text);

cmd.Parameters.AddWithValue("phone", this.tbPhone.Text);

cmd.Parameters.AddWithValue("address", this.tbAddress.Text);

int id = int.Parse(dataGridView1.CurrentRow.Cells[0].Value.ToString());

cmd.Parameters.AddWithValue("id", id);

cmd.Prepare();

cmd.ExecuteNonQuery();

loadClients();

this.tbLastname.Text = "";

this.tbName.Text = "";

this.tbSurname.Text = "";

this.tbPassport.Text = "";

this.tbPhone.Text = "";

this.tbAddress.Text = "";

}

Код обработчика кнопки для удаления записей из таблицы с информацией о клиентах:

private void btnClientsDelete\_Click(object sender, EventArgs e)

{

string sql = $"DELETE FROM clients WHERE id=(@id)";

NpgsqlCommand cmd = new NpgsqlCommand(sql, con);

int id = int.Parse(dgvClients.CurrentRow.Cells[0].Value.ToString());

cmd.Parameters.AddWithValue("id", id);

cmd.Prepare();

cmd.ExecuteNonQuery();

loadClients();

}

Аналогичным образом осуществляется добавление, удаление и изменение записей в таблицах с информацией о товарах и договорах.

Работу программы можно рассмотреть на примере изменения записи в таблице с информацией о клиентах. При нажатии кнопки “Изменение” на главной форме открывается диалоговое окно с формой для заполнения информации, представленное на рисунке 2.

Изображение выглядит как текст, снимок экрана, программное обеспечение, дисплей

Автоматически созданное описание

Рисунок 2 − окно для изменения записи в таблице

При выборе строки таблицы форма заполняется информацией о выбранном клиенте для изменения (рисунок 3).

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

Рисунок 3 − заполнение формы

После изменения данных и нажатия на кнопку “Изменить” таблица с информацией о клиентах обновляется.

Ссылка на полный проект на github: <https://github.com/MostovayaE/RPvIS>.

**Вывод:** таким образом, в соответствии с заданием, мною разработано клиентское приложение на языке С# для работы с базой данных формата PostgreSQL.