Лабораторная работа №2

**ЗНАКОМСТВО С СУБД *MongoDB***

**Цель работы:** установить СУБД *MongoDB* и настроить связь со средой разработки *Visual Studio.*

**Теоретические основы**

*MongoDB* — это высокопроизводительная документо-ориентированная база данных. *MongoDB* реализует новый подход к построению баз данных, где нет таблиц, схем, запросов *SQL*, внешних ключей и прочего, что присуще объектно-реляционным базам данных.

Одним из популярных стандартов обмена данными и их хранения является *JSON (JavaScript Object Notation). JSON* эффективно описывает сложные по структуре данные. Для хранения данных в *MongoDB* применяется формат, который называется *BSON*. Данный формат позволяет быстрее выполнять поиск данных и их обработку.

*MongoDB* написана на *C++,* поэтому с ней можно работать на самых разных платформах. *MongoDB* может быть развернута на платформах *Windows, Linux, MacOS, Solaris*

**Выполнение работы**

В ходе выполнения работы была установлена СУБД *MongoDB,* создана коллекция *Employee*, содержащая информацию о сотрудниках (*ID, Name, Salary*) и создано *WPF* приложение на языке *С#,* на примере которого продемонстрирована связь базы данных со средой разработки *Visual Studio 2019.*

Данное приложение позволяет увидеть содержимое коллекции сотрудников (*Employee*), а также вставлять, изменять и удалять данные из этой коллекции. Главный экран приложения продемонстрирован на рисунке 1

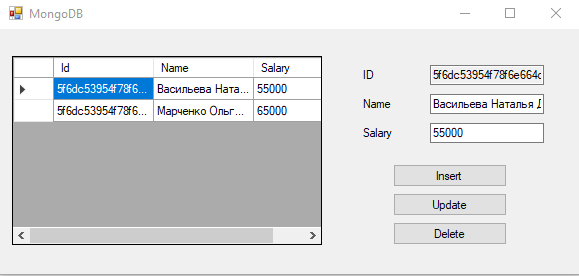


Рисунок 1 – Главный экран приложения

Для того, чтобы добавить новую запись, необходимо заполнить поля «*Name»* и «*Salary»* и нажать на кнопку «*Insert*».

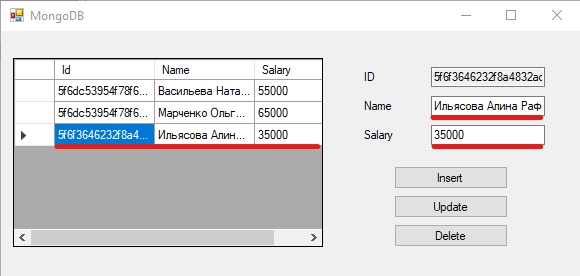


Рисунок 2 - Результат добавления новой записи

Для того, чтобы изменить данные существующей записи, необходимо выбрать ее из таблицы, изменить значение поля «*Name»* и/или «*Salary»* и нажать на кнопку «*Update*».

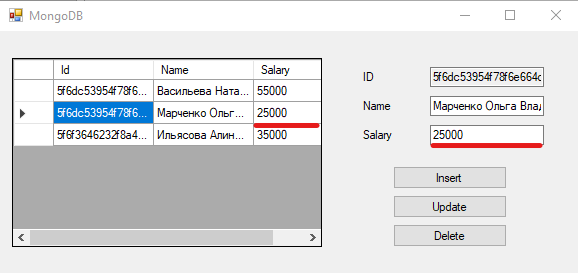


Рисунок 3 – Результат изменения данных

Для удаления записи необходимо выбрать ее из таблицы и нажать на кнопку «*Delete*».

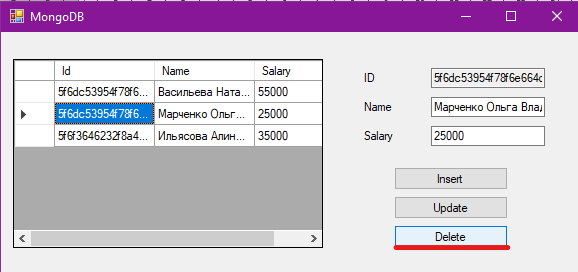


Рисунок 4 – Удаление записи из коллекции

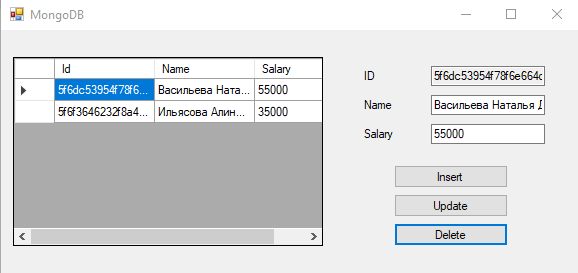


Рисунок 5 – Результат удаления записи из коллекции

**Вывод**: в ходе выполнения лабораторной работы была установлена СУБД *MongoDB* и настроена связь со средой разработки *Visual Studio.* Также было создано *WPF* приложение, демонстрирующее связь базы данных и среды разработки на примере изменения данных коллекции.

**Листинг:**

**Employee.cs**

using MongoDB.Bson;

using MongoDB.Bson.Serialization.Attributes;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace WF\_MongoDB

{

class Employee

{

[BsonId]

public ObjectId Id { get; set; }

[BsonElement("name")]

public String Name { get; set; }

[BsonElement("salary")]

public double Salary { get; set; }

public Employee(string name, double salary)

{

Name = name;

Salary = salary;

}

}

}

**Form1.cs**

using MongoDB.Bson;

using MongoDB.Driver;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace WF\_MongoDB

{

public partial class Form1 : Form

{

static MongoClient client = new MongoClient();

static IMongoDatabase db = client.GetDatabase("employeeDB");

static IMongoCollection<Employee> collection = db.GetCollection<Employee>("employee");

public void ReadAllDocuments()

{

List<Employee> list = collection.AsQueryable().ToList<Employee>();

dataGridView1.DataSource = list;

textBox1.Text = dataGridView1.Rows[0].Cells[0].Value.ToString();

textBox2.Text = dataGridView1.Rows[0].Cells[1].Value.ToString();

textBox3.Text = dataGridView1.Rows[0].Cells[2].Value.ToString();

}

public Form1()

{

InitializeComponent();

ReadAllDocuments();

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e)

{

textBox1.Text = dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString();

textBox2.Text = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();

textBox3.Text = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();

}

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

{

Employee emp = new Employee(textBox2.Text, Double.Parse(textBox3.Text));

collection.InsertOne(emp);

ReadAllDocuments();

}

private void button2\_Click(object sender, EventArgs e)

{

var updateDef = Builders<Employee>.Update.Set("name", textBox2.Text).Set("salary", textBox3.Text);

collection.UpdateOne(emp => emp.Id == ObjectId.Parse(textBox1.Text), updateDef);

ReadAllDocuments();

}

private void button3\_Click(object sender, EventArgs e)

{

collection.DeleteOne(emp => emp.Id == ObjectId.Parse(textBox1.Text));

ReadAllDocuments();

}

}

}