

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
«Владимирский государственный университет
Имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИВлГУ)

Факультет _____ ИТР _____

Кафедра _____ ПИН _____

ЛАБОРАТОРНАЯ РАБОТА №1

по _____ Системы управления базами данных _____
Тема _____ Работа с СУБД SQLITE. _____

Руководитель

_____ Колпаков А.А. _____

(фамилия, инициалы)

_____ (подпись)

_____ (дата)

Студент _____ ПИН-122 _____

(группа)

_____ Хафизова М.М. _____

(фамилия, инициалы)

_____ (подпись)

_____ (дата)

Муром 2024

Лабораторная работа №1

Цель работы: Изучить основы СУБД SQLITE.

Задание:

Вариант 21

Создать программу для работы с базой данных. Создать таблицу. Добавить в таблицу записи, редактировать эти записи и очистить таблицу.

					МИВУ 09.03.04-01.21			
Изм.	Лист	№ докум.	Подп.	Дата	Работа с СУБД SQLITE.			
Разраб.		Хафизова М.М.						
Провер.		Колпаков А.А.						
Н.контр.								
Утв.					МИВУ ПИН - 122			
						Лит.	Лист	Листов
							2	7

Ход работы:

Листинг 1:

```
namespace sqlite
{
    public partial class Form1 : Form
    {
        private sqliteclass mydb = null;
        private string sCurDir = string.Empty;
        private string sPath = string.Empty;
        private string sSql = string.Empty;

        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            sPath = Path.Combine(Application.StartupPath, "mybd.db");
            Text = sPath;
        }

        private void button1_Click(object sender, EventArgs e)
        {
            mydb = new sqliteclass();
            sSql = @"CREATE TABLE if not exists [musician]([id] INTEGER PRIMARY KEY
AUTOINCREMENT, [FIO] TEXT NOT NULL,[lastPerformance] datetime NOT NULL,[nickname]
INTEGER DEFAULT 0);";
            //Пытаемся создать таблицу
            mydb.iExecuteNonQuery(sPath, sSql, 0);
            //Проверка работы
            Text = "Таблица создана!";
            mydb = null;
            return;
        }
        private void button2_Click(object sender, EventArgs e)
        {
            mydb = new sqliteclass();
            sSql = @"insert into musician (FIO,lastPerformance,nickname)
values('Михаил Юрьевич Горшенёв','2004-07-08',1);";
            //Проверка работы
            if (mydb.iExecuteNonQuery(sPath, sSql, 1) == 0)
            {
                Text = "Ошибка записи!";
            }
            mydb = null;
            Text = "Запись 1 добавлена!";
            return;
        }
        private void button3_Click(object sender, EventArgs e)
        {
            mydb = new sqliteclass();
            //Ошибка в дате намеренная
        }
    }
}
```

Изм.	Лист	№ докум.	Подп.	Дата

МИВУ 09.03.04-01.10

Лист

3

```

        sSql = @"insert into musician (FIO,lastPerformance,nickname)
values('Князев Андрей Сергеевич','2004-03-12',2)";
        //Проверка работы
        if (mydb.iExecuteNonQuery(sPath, sSql, 1) == 0)
        {
            Text = "Ошибка записи!";
        }
        mydb = null;
        Text = "Запись 2 добавлена!";

    }

    private void button4_Click(object sender, EventArgs e)
    {
        mydb = new sqliteclass();
        sSql = "select * from musician";
        DataRow[] datarows = mydb.drExecute(sPath, sSql);
        if (datarows == null)
        {
            Text = "Ошибка чтения!";
            mydb = null;
            return;
        }
        Text = "";
        foreach (DataRow dr in datarows)
        {
            Text += dr["id"].ToString().Trim() + " " +
            dr["FIO"].ToString().Trim()
            + " " + dr["lastPerformance"].ToString().Trim() + " " +
            dr["nickname"].ToString().Trim() + " ";
        }
        mydb = null;
    }

    private void button5_Click(object sender, EventArgs e)
    {
        mydb = new sqliteclass();
        sSql = "delete from musician";
        if (mydb.iExecuteNonQuery(sPath, sSql, 1) == 0)
        {
            Text = "Ошибка удаления записи!";
            mydb = null;
            return;
        }
        mydb = null;
        Text = "Записи удалены из БД!";
        return;
    }

    private void button6_Click(object sender, EventArgs e)
    {
        mydb = new sqliteclass();
        sSql = @"Update musician set lastPerformance='2017-03-05' where FIO
like('%Михаил Юрьевич Горшенёв%')";
        //Проверка работы
        if (mydb.iExecuteNonQuery(sPath, sSql, 1) == 0)
        {
            Text = "Ошибка обновления записи!";

```

Изм.	Лист	№ докум.	Подп.	Дата

```

        mydb = null;
        return;
    }
    mydb = null;
    Text = "Запись 1 исправлена!";
}

}
}

```

```

namespace sqlite
{
    internal class sqliteclass
    {
        public sqliteclass()
        {
        }
        #region ExecuteNonQuery
        public int iExecuteNonQuery(string FileData, string sSql, int where)
        {
            int n = 0;
            try
            {
                using (SQLiteConnection con = new SQLiteConnection())
                {
                    if (where == 0)
                    {
                        con.ConnectionString = @"Data Source=" + FileData +
                            ";New=True;Version=3";
                    }
                    else
                    {
                        con.ConnectionString = @"Data Source=" + FileData +
                            ";New=False;Version=3";
                    }
                    con.Open();
                    using (SQLiteCommand sqlCommand = con.CreateCommand())
                    {
                        sqlCommand.CommandText = sSql;
                        n = sqlCommand.ExecuteNonQuery();
                    }
                    con.Close();
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
                n = 0;
            }
            return n;
        }
        #endregion
        #region Execute
        public DataRow[] drExecute(string FileData, string sSql)
        {
            DataRow[] datarows = null;

```

Изм.	Лист	№ докум.	Подп.	Дата

МИВУ 09.03.04-01.10

Лист

5

```

        SQLiteDataAdapter dataadapter = null;
        DataSet dataset = new DataSet();
        DataTable datatable = new DataTable();
        try
        {
            using (SQLiteConnection con = new SQLiteConnection())
            {
                con.ConnectionString = @"Data Source=" + FileData +
                ";New=False;Version=3";
                con.Open();
                using (SQLiteCommand sqlCommand = con.CreateCommand())
                {
                    dataadapter = new SQLiteDataAdapter(sSql, con);
                    dataset.Reset();
                    dataadapter.Fill(dataset);
                    datatable = dataset.Tables[0];
                    datarows = datatable.Select();
                }
                con.Close();
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
            datarows = null;
        }
        return datarows;
    }
#endregion
}

```




Рисунок 1 – Создание таблицы



Рисунок 2 – Добавление 1 записи



Рисунок 3 – Добавленик 2 записи


1 Михаил Юрьевич Горшенёв 08.07.2004 0:00:00 1 2 Князев Андрей Сергеевич 12.03.2004 0:00:00 2

Create

Insert 1


Insert 2

Read

Delete

Update

Рисунок 4 – Чтение записей из таблицы


Запись 1 исправлена!

Create

Insert 1


Insert 2

Read

Delete

Update

Рисунок 5 – Исправление записей


1 Михаил Юрьевич Горшенёв 05.03.2017 0:00:00 1 2 Князев Андрей Сергеевич 12.03.2004 0:00:00 2

Create

Insert 1


Insert 2

Read

Delete

Update

Рисунок 6 – Чтение исправленных записей из таблицы


Записи удалены из БД!

Create

Insert 1

Insert 2

Read

Delete

Update

Рисунок 7 – Удаление записей из таблицы

Вывод: В ходе выполнения данной лабораторной работы были изучены основы СУБД SQLITE.