**Задание**

1. Изучить принципы физической организации баз данных
2. Изучить организацию файлов, файловой системы, методов работы с файлами в программировании
3. Изучить организацию схемы базы данных, структуры таблицы, типы данных в таблицах
4. Изучить понятия индекса и ключевого поля таблицы, операции индексирования, составного индекса
5. Создать записную книжку обычного человека, который имеет относительно мало знакомых.
6. Для этого создать приложение на языке C, C ++ или C #. Таблица БД должна быть размещена в файле и представлять собой массив структур (по терминологии С)
7. Для каждого абонента надо хранить такие данные (структуру): 1. ФИО (фамилия, имя и отчество) абонента; 2.Номер стационарного телефона; 3.Номер мобильного телефона; 4.Електроный адрес; 5.Категория абонента; 6.Домашний адрес; 7. Примечание

(Под *Категорией* абонента будем понимать обновляемый список: одноклассник, сотрудник, знакомый и т. д. Очевидно, что некоторые поля записи будут не заполнены. Но поле ФИО обязательно должно быть заполнено. Все поля, касающиеся телефона, адреса или электронный адрес не могут быть пустыми одновременно)

1. Назвать БД "Записная книжка", она состоит только из одной таблицы - "Телефоны". Необходимо выбрать названия и атрибуты полей таблицы.

|  |  |  |
| --- | --- | --- |
| Название поля | Тип поля | Примечания |
| ФИО | текст | всегда присутствует |
| НомСтацТелеф | целое | 2358974 |
| НомМобилТелеф | целое | 0675623487 |
| Дата рождения | дата | 01.01.1990 |
| Электронный адрес | текст | qqqqqqqqqqq@qqqqqqqqq.qq |
| Категория\_абонента | текст | сотрудник |
| Домашний адрес | текст | ул. Йййййййййй, д.99, кв. 99 |
| Примечание | текст |  |

1. Разработать приложение, которое должно обеспечить следующие функции:

* добавление нового абонента
* редактирование существующего абонента
* удаление существующего абонента
* последовательный поиск абонента по одному из параметров (значению поля, заданного целиком)

1. Значения для вставки и редактирования поля «Категория\_абонента» должны храниться и выбираться из отдельного файла
2. Для файла значений поля «Категория\_абонента» обеспечить следующие операции:

* добавления нового элемента списка;
* редактирования существующего элемента списка;
* удаление существующего элемента списка.

**Отчет**

1. Основные сведения о физической организации баз данных
   1. Организация (размещение) данных во внутри машинной сфере может рассматриваться на двух уровнях — физическом и логическом. Физическая организация данных определяет способ размещения данных непосредственно на машинном носителе. В современных прикладных программных средствах БД этот уровень организации обеспечивается автоматически без вмешательства пользователя.

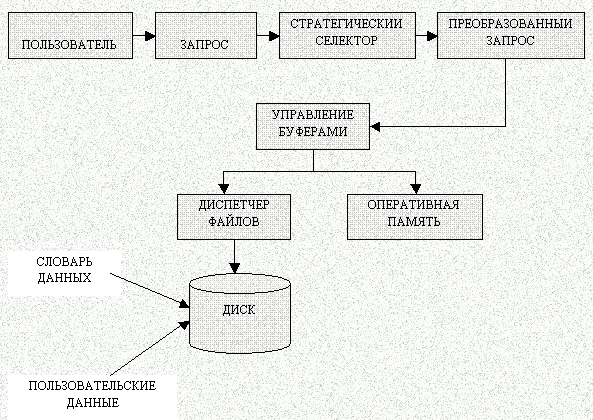


Рис. 1. Система физического доступа к БД.

* 1. Стратегический селектор – это программа, которая преобразует запрос в эффективную для исполнения форму.
  2. Программа управления буферами – контролирует обмен информации между оперативной памятью и диском.
  3. Диспетчер файлов – программа, управляющая размещением файлов на диске.
  4. Словарь данных – это часть СУБД, определяющая структуру пользовательских данных и возможности их использования.

1. Основные сведения об организации файлов, файловой системы, о методах работы с файлами в программировании.
   1. Файловая система - это часть операционной системы, назначение которой состоит в том, чтобы обеспечить пользователю удобный интерфейс при работе с данными, хранящимися на диске, и обеспечить совместное использование файлов несколькими пользователями и процессами. Каждый диск разбивается на две области: область хранения файлов и каталог. Каталог содержит имя файла и указание на начало его размещения на диске. Если провести аналогию диска с книгой, то область хранения файлов соответствует ее содержанию, а каталог - оглавлению. Причем книга состоит из страниц, а диск - из секторов.
   2. Подобно паре Directory/DirectoryInfo для работы с файлами предназначена пара классов File и FileInfo. С их помощью мы можем создавать, удалять, перемещать файлы, получать их свойства и многое другое.
   3. Класс FileStream представляет возможности по считыванию из файла и записи в файл. Он позволяет работать как с текстовыми файлами, так и с бинарными.
   4. Класс FileStream не очень удобно применять для работы с текстовыми файлами. К тому же для этого в пространстве System.IO определены специальные классы: StreamReader и StreamWriter.
2. Основные сведения о структуре таблиц, типах данных в таблицах.
   1. В Access данные упорядочиваются в таблицах, которые представляют собой списки строк и столбцов, напоминающие бухгалтерский блокнот или электронную таблицу. В простой базе данных может быть всего одна таблица. Для большинства баз данных их потребуется несколько. Например, в одной таблице можно хранить сведения о товарах, в другой — о заказах, а в третьей — о клиентах. Каждую строку правильнее называть записью, а каждый столбец — полем. Запись — это эффективный и согласованный способ объединения сведений о чем-либо. Поле — это отдельный элемент сведений (элементы такого типа есть в каждой записи).
3. Скриншоты экрана приложения с результатом работы программы.

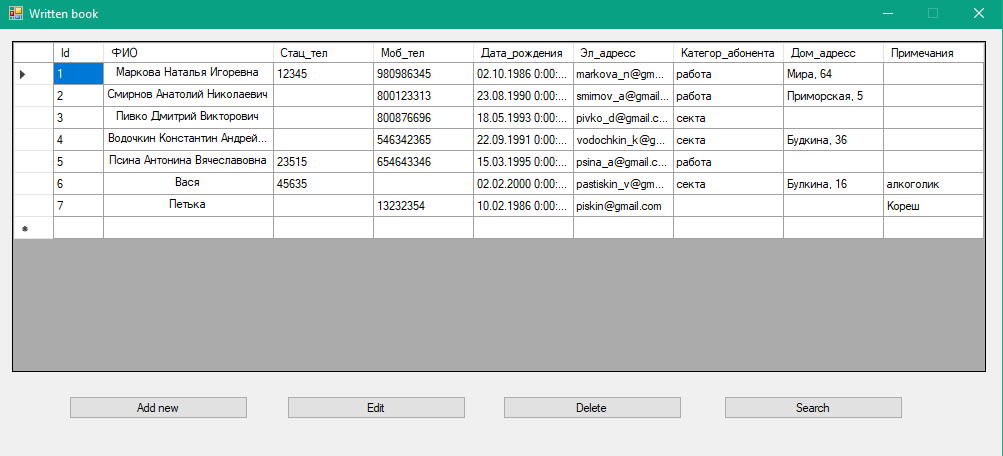


Рис. 7. Интерфейс программы.

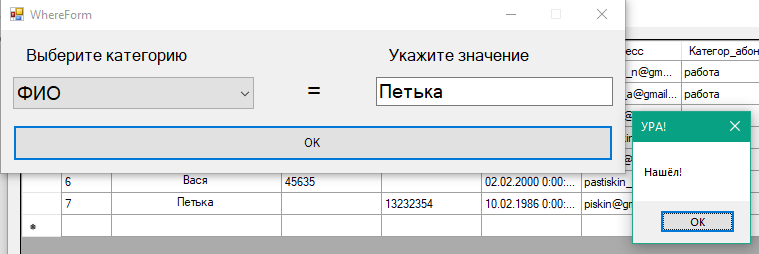


Рис. 8. Поиск элементов.

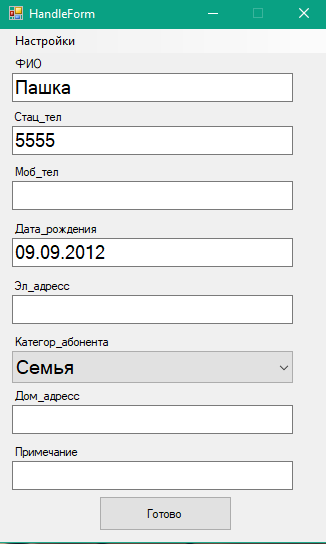


Рис. 9. Добавление нового контакта.



Рис. 10. Результат добавления.

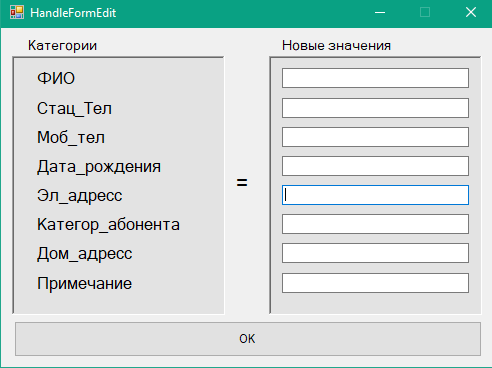


Рис. 11. Форма для редактирования контактов.

1. Текст исходных кодов программы.
   1. Главная форма.

using System;

using System.Windows.Forms;

using System.Data.OleDb;

namespace DB\_1

{

public partial class Form1 : Form

{

private OleDbConnection connection = new OleDbConnection("Provider = Microsoft.Jet.OLEDB.4.0; Data Source = Writen\_book.mdb");

private string name, phone\_numb, mob\_phone\_numb, birthday, email, peop\_type, adress, remark;

HandleFormAdd hf;

public Form1()

{

InitializeComponent();

FormLoad();

}

private void SearchBtnClick(object sender, EventArgs e)

{

string query = "SELECT [ФИО] FROM [телефоны] ";

using (WhereForm wf = new WhereForm(this, query, ref connection))

{

wf.ShowDialog();

if (wf.DialogResult == DialogResult.OK)

wf.Close();

}

}

private void AddBtnClick(object sender, EventArgs e)

{

string query = null;

int ID = 1;

using (OleDbCommand command = new OleDbCommand("SELECT MAX([Код]) FROM телефоны ", connection))

{

OleDbDataReader reader = command.ExecuteReader();

reader.Read();

ID += int.Parse(reader[0].ToString());

reader.Close();

}

using (hf = new HandleFormAdd(this))

{

hf.ShowDialog();

if (hf.DialogResult == DialogResult.OK)

{

hf.Close();

if (phone\_numb == "")

{

query = $"INSERT INTO [телефоны] VALUES ('{ID}', '{name}', NULL, '{int.Parse(mob\_phone\_numb)}', {birthday}, '{email}', '{peop\_type}', '{adress}', '{remark}'); ";

}

else if (mob\_phone\_numb == "")

{

query = $"INSERT INTO [телефоны] VALUES ('{ID}', '{name}', '{int.Parse(phone\_numb)}', NULL, {birthday}, '{email}', '{peop\_type}', '{adress}', '{remark}'); ";

}

else

{

query = $"INSERT INTO [телефоны] VALUES ('{ID}', '{name}', '{int.Parse(phone\_numb)}', '{int.Parse(mob\_phone\_numb)}', {birthday}, '{email}', '{peop\_type}', '{adress}', '{remark}'); ";

}

try

{

using (OleDbCommand command = new OleDbCommand(query, connection))

{

int i = command.ExecuteNonQuery();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

DataGridLoad();

}

private void EditBtnClick(object sender, EventArgs e)

{

using (HandleFormEdit.HandleFormEdit hfe = new HandleFormEdit.HandleFormEdit(this, ref connection))

{

hfe.ShowDialog();

if (hfe.DialogResult == DialogResult.OK)

{

hfe.Close();

DataGridLoad();

}

}

}

private void DeleteBtnClick(object sender, EventArgs e)

{

string query = "DELETE FROM [телефоны] ";

using (WhereForm wf = new WhereForm(this, query, ref connection))

{

wf.ShowDialog();

if (wf.DialogResult == DialogResult.OK)

wf.Close();

}

DataGridLoad();

}

public void SetData(string name, string phone\_numb, string mob\_phone\_numb, string birthday, string email, string peop\_type, string adress, string remark)

{

this.name = name;

this.phone\_numb = phone\_numb;

this.mob\_phone\_numb = mob\_phone\_numb;

this.birthday = birthday;

NullCheck(ref this.birthday);

this.email = email;

this.peop\_type = peop\_type;

this.adress = adress;

this.remark = remark;

}

private void NullCheck(ref string value)

{

string temp = null;

if (value == "")

value = "NULL";

else

{

temp += "\'" + value + "\'";

value = temp;

}

}

private void FormLoad()

{

connection.Open();

DataGridLoad();

}

private void DataGridLoad()

{

dataGridView1.Rows.Clear();

using (OleDbCommand command = new OleDbCommand("SELECT \* FROM [телефоны]", connection))

{

OleDbDataReader reader = command.ExecuteReader();

while (reader.Read())

dataGridView1.Rows.Add(reader[0], reader[1], reader[2], reader[3], reader[4], reader[5], reader[6], reader[7], reader[8]);

reader.Close();

}

}

private void FormClose(object sender, FormClosingEventArgs e)

{

connection.Close();

}

}

}

6.2. HandleFormAdd.cs

using System;

using System.Windows.Forms;

using System.IO;

namespace DB\_1

{

public partial class HandleFormAdd : Form

{

Form1 f1;

public HandleFormAdd()

{

InitializeComponent();

FormLoad();

}

public HandleFormAdd(Form1 form1)

{

InitializeComponent();

FormLoad();

f1 = form1;

f1.Enabled = false;

}

private void EditUserClick(object sender, EventArgs e)

{

Editor editor = new Editor(this);

editor.ShowDialog();

if (editor.DialogResult == DialogResult.OK)

{

peopType\_comboBox.Items.Clear();

using (StreamReader reader = new StreamReader("categories.txt"))

{

string temp = null;

peopType\_comboBox.Items.Add("\n");

while ((temp = reader.ReadLine()) != null)

peopType\_comboBox.Items.Add(temp);

}

}

}

private void KeyPressCheck(object sender, KeyPressEventArgs e)

{

char symb = e.KeyChar;

if (!char.IsDigit(symb) && symb != 8 && symb != 46)

e.Handled = true;

}

private void DoneBtnClick(object sener, EventArgs e)

{

if (name\_textBox.Text == "")

MessageBox.Show("Укажите ФИО!", "Warning!");

else if (phoneNum\_textBox.Text == "" && mobPhoneNum\_textBox.Text == "" && email\_textBox.Text == "" && adress\_textBox.Text == "")

MessageBox.Show("Укажите что-то из этого: стационарный/мобильный номер, электронный/домашний адрес.", "Warning!");

else

{

f1.SetData(name\_textBox.Text, phoneNum\_textBox.Text, mobPhoneNum\_textBox.Text, birthday\_textBox.Text, email\_textBox.Text, peopType\_comboBox.Text, adress\_textBox.Text, remark\_textBox.Text);

DialogResult = DialogResult.OK;

Hide();

}

}

private void FormLoad()

{

try

{

using (StreamReader reader = new StreamReader("categories.txt"))

{

string temp = null;

peopType\_comboBox.Items.Add("\n");

while ((temp = reader.ReadLine()) != null)

peopType\_comboBox.Items.Add(temp);

}

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void FormClose(object sender, FormClosingEventArgs e)

{

f1.Enabled = true;

}

}

}

6.3. Editor.cs

using System;

using System.Windows.Forms;

using System.IO;

namespace DB\_1

{

public partial class Editor : Form

{

HandleFormAdd hf;

public Editor()

{

InitializeComponent();

FormLoad();

}

public Editor(HandleFormAdd handleForm)

{

InitializeComponent();

FormLoad();

hf = handleForm;

hf.Enabled = false;

}

private void OkBtnClick(object sender, EventArgs e)

{

using (StreamWriter writer = new StreamWriter("categories.txt", false))

{

string[] temp = richTextBox.Text.Split('\n');

for(int i = 0; i < temp.Length; i++)

writer.WriteLine(temp[i]);

}

DialogResult = DialogResult.OK;

Close();

}

private void FormLoad()

{

using (StreamReader reader = new StreamReader("categories.txt"))

{

string temp = null;

while ((temp = reader.ReadLine()) != null)

richTextBox.Text += temp + "\n";

}

}

private void FormClose(object sender, FormClosingEventArgs e)

{

hf.Enabled = true;

}

}

}

6.4. HandleFormEdit.cs

using System;

using System.Windows.Forms;

using System.Data.OleDb;

using System.Collections.Generic;

namespace DB\_1.HandleFormEdit

{

public partial class HandleFormEdit : Form

{

private Form1 f1;

private OleDbConnection connection;

List<Label> tempTxt;

List<TextBox> tempVal;

public HandleFormEdit()

{

InitializeComponent();

}

public HandleFormEdit(Form1 f1, ref OleDbConnection connection)

{

InitializeComponent();

this.f1 = f1;

this.connection = connection;

f1.Enabled = false;

}

private void OkBtnClick(object sender, EventArgs e)

{

DialogResult = DialogResult.OK;

tempTxt = new List<Label>() { name\_txt, phone\_txt, mob\_phone\_txt, birthday\_txt, email\_txt, category\_txt, adress\_txt, remark\_txt};

tempVal = new List<TextBox>() { name\_val, phone\_val, mob\_phone\_val, birthday\_val, email\_val, category\_val, adress\_val, remark\_val};

string query = "UPDATE [телефоны] SET ";

GetData(ref query, tempTxt, tempVal);

using (WhereForm wf = new WhereForm(this, query, ref connection))

{

wf.ShowDialog();

if (wf.DialogResult == DialogResult.OK)

wf.Close();

}

Hide();

}

private void GetData(ref string query, List<Label> tempTxt, List<TextBox> tempVal)

{

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

{

if (tempVal[i].Text != "")

{

if (tempTxt[i].Text == "Стац\_Тел" || tempTxt[i].Text == "Моб\_тел")

query += $"[{tempTxt[i].Text}] = {int.Parse(tempVal[i].Text)},";

else if (tempTxt[i].Text == "Дата\_рождения")

{

string[] temp;

string str = "s";

tempVal[i].Text = tempVal[i].Text.Replace('.', '/');

temp = tempVal[i].Text.Split('/');

tempVal[i].Text = tempVal[i].Text.Replace(temp[0], str);

tempVal[i].Text = tempVal[i].Text.Replace(temp[1], temp[0]);

tempVal[i].Text = tempVal[i].Text.Replace(str, temp[1]);

query += $"[{tempTxt[i].Text}] = '{tempVal[i].Text}',";

}

else

query += $"[{tempTxt[i].Text}] = '{tempVal[i].Text}',";

}

}

if (query[query.Length-1] == ',')

{

query = query.Substring(0, query.Length -1);

}

}

private void KeyPressCheck(object sender, KeyPressEventArgs e)

{

char symb = e.KeyChar;

if (!char.IsDigit(symb) && symb != 8 && symb != 46)

e.Handled = true;

}

private void FormClose(object sender, FormClosingEventArgs e)

{

f1.Enabled = true;

}

}

}

6.5. WhereForm.cs

using System;

using System.Data.OleDb;

using System.Windows.Forms;

namespace DB\_1

{

public partial class WhereForm : Form

{

private HandleFormEdit.HandleFormEdit hfe = null;

private Form1 f1 = null;

private string query = null;

private OleDbConnection connection;

public WhereForm()

{

InitializeComponent();

}

public WhereForm(HandleFormEdit.HandleFormEdit hfe, string query, ref OleDbConnection connection)

{

InitializeComponent();

this.hfe = hfe;

this.query = query;

this.connection = connection;

hfe.Enabled = false;

}

public WhereForm(Form1 f1, string query, ref OleDbConnection connection)

{

InitializeComponent();

this.f1 = f1;

this.query = query;

this.connection = connection;

f1.Enabled = false;

}

private void OkBtnClick(object sender, EventArgs e)

{

string[] temp = null;

string tempStr = "str";

DialogResult = DialogResult.OK;

if (cmbBox.Text == "Код" || cmbBox.Text == "Стац\_Тел" || cmbBox.Text == "Моб\_тел")

query += $"WHERE [{cmbBox.Text}] = {int.Parse(txtBox.Text)};";

else if (cmbBox.Text == "Дата\_рождения")

{

txtBox.Text = txtBox.Text.Replace('.', '/');

temp = txtBox.Text.Split('/');

txtBox.Text = txtBox.Text.Replace(temp[0], tempStr);

txtBox.Text = txtBox.Text.Replace(temp[1], temp[0]);

txtBox.Text = txtBox.Text.Replace(tempStr, temp[1]);

query += $"WHERE (((телефоны.[{cmbBox.Text}]) = #{txtBox.Text}#));";

}

else

query += $"WHERE [{cmbBox.Text}] = '{txtBox.Text}';";

try

{

temp = query.Split(' ');

using (OleDbCommand command = new OleDbCommand(query, connection))

{

if (temp[0] == "SELECT")

{

OleDbDataReader reader = command.ExecuteReader();

reader.Read();

MessageBox.Show("Нашёл!", "УРА!");

}

else

{

command.ExecuteNonQuery();

}

}

}

catch(Exception ex)

{

MessageBox.Show(ex.Message);

}

Hide();

}

private void KeyPressCheck(object senser, KeyPressEventArgs e)

{

char symb = e.KeyChar;

if (cmbBox.Text == "Код" || cmbBox.Text == "Стац\_Тел" || cmbBox.Text == "Моб\_тел" || cmbBox.Text == "Дата\_рождения")

{

if (!char.IsDigit(symb) && symb != 8 && symb != 46)

e.Handled = true;

}

}

private void FormClose(object sender, FormClosingEventArgs e)

{

if(hfe != null)

hfe.Enabled = true;

if (f1 != null)

f1.Enabled = true;

}

}

}

1. Краткие выводы.

Изучил принципы физической организации баз данных. Изучил организацию файлов, файловой системы, методов работы с файлами в программировании. Изучил организацию схемы базы данных, структуры таблицы, типы данных в таблицах. Изучил понятия индекса и ключевого поля таблицы, операции индексирования, составного индекса.

1. Перечень использованных источников
   1. <https://metanit.com/sharp/tutorial/5.5.php>
   2. <https://metanit.com/sharp/tutorial/5.3.php>
   3. <https://siblec.ru/informatika-i-vychislitelnaya-tekhnika/bazy-i-banki-dannykh/11-fizicheskaya-organizatsiya-bd>
   4. <https://sites.google.com/a/i-dist.ru/informacionnye-tehnologii-ucebnoe-posobie/operacionnye-sistemy-personalnogo-komputera/fajly-i-fajlovaa-sistema>
   5. <http://citforum.ck.ua/operating_systems/sos/glava_10.shtml>