МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Криворізький національний університет

Кафедра моделювання і програмного забезпечення

**ЛАБОРАТОРНА РОБОТА №2**

з дисципліни «Базы данных»

тема «Разработка интерактивного конструктора запросов.

(создание, хранение и использование запросов (основные типы))»

Виконав:

студент групи ІПЗ-18-2

Перемєна Д. А.

Перевірили:

Доценко І. О.

Білашенко С. В.

Кривий Ріг

2020

**Задание**

1. Изучить принципы физической организации баз данных
2. Изучить организацию схемы и структуры базы данных, структуры таблиц, типов данных в таблицах
3. Изучить основные типы запросов в базах данных – на создание БД, на создание таблиц, на выборку, на добавление, на удаление и на модификацию данных
4. Изучить интегрированные среды разработки ПО и программные средства управления базами данных
5. Разработать схему базу данных в виде UML-диаграммы: таблица *КАФЕДРЫ(поля: название, факультет, заведующий);* *СТУДЕНТЫ(поля: фамилия, группа, год\_поступления, рейтинг(от 0,0 до 5,0 баллов); ГРУППЫ(поля: шифр\_группы, специальность, кафедра); СПЕЦИАЛЬНОСТИ(поля: код, название, отрасль знаний)).* Дополнительно предусмотреть в таблицах индексные и ключевые поля для связи таблиц и нормализации базы данных
6. Разработать программу для интерактивной генерации текстов запросов основных типов (см. п. 3)
7. Создать основную форму программы. На форме расположить элементы управления – выпадающие списки (*ComboBox*) для возможности выбора имен таблиц и полей, перечисленных в схеме базы данных (см. п.5)
8. На форме расположить элемент управления – выпадающий список (типа *ComboBox*) с выбором типов запросов (см. п. 3)
9. Предусмотреть возможность интерактивного пополнения любого выпадающего списка новыми значениями с их сортировкой, обновлением и отображением
10. На форме расположить элемент управления – многострочное текстовое поле (типа *Memo*) для хранения текста результирующего запроса. Вначале сделать его невидимым
11. На форме расположить элемент управления – кнопку (типа *Button*) для формирования текста запроса
12. На форме расположить главное меню с пунктом: «Файл» и подпунктами «Создать», «Сохранить», «Открыть», «Выйти»
13. На форме расположить элементы управления – «стандартные диалоги» (типа *Dialog*). С их помощью реализовывать функции главного меню (см. п. 12)
14. Реализовать следующие функции программы:
    1. Выбор типа запроса (см. п. 3)
    2. Выбор таблиц для запроса
    3. Выбор полей для таблицы (при выборе конкретной таблицы в одном выпадающем списке в выпадающем списке полей должны отображаться только поля выбранной таблицы)
    4. При нажатии на кнопку формировать текст SQL-запроса и записывать его в многострочное текстовое поле (типа *Memo*)
    5. Реализовать функции главного меню: «Создать» (текстовое поле запроса становится видимым), «Сохранить» (текст построенного запроса сохраняется в файл), «Открыть» (текст запроса загружается из файла в текстовое поле), «Выйти» (закрывается программа, если поле содержит текст запроса, - то предложить его сохранить в файл)
15. С помощью разработанного конструктора сформировать и сохранить в отдельных файлах тексты разработанных запросов
16. В локальной СУБД (например, – MS Access) создать базу данных на основе схемы БД (модели - см. п. 5)
17. Сформировать запросы на основе текстов запросов, созданных с помощью своего разработанного интерактивного конструктора запросов. Проверить их работоспособность.
18. Зафиксировать ошибки в запросах. Сформировать их снова в своем конструкторе и снова проверить. Добиться работоспособности всех запросов. Сделать выводы

**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.

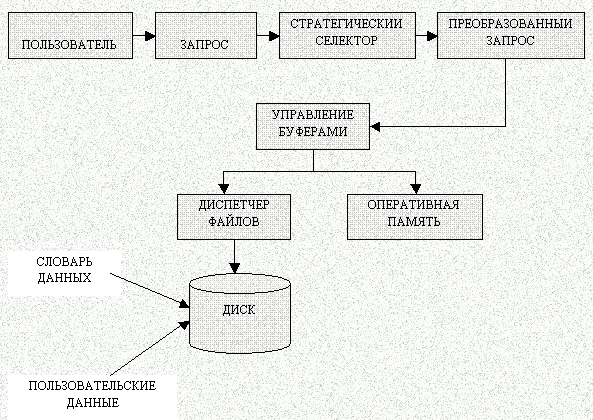


Рис. . Система физического доступа к БД.

* 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 основных типах запросов.

Выделяют такие виды SQL запросов:

* 1. DDL (Data Definition Language) - язык определения данных. Задачей DDL запросов является создание БД и описание ее структуры. Запросами такого вида устанавливаются правила того, в каком виде различные данные будут размещаться в БД.
  2. DML (Data Manipulation Language) - язык манипулирования данными. В число запросов этого типа входят различные команды, используя которые непосредственно производятся некоторые манипуляции с данными. DML-запросы нужны для добавления изменений в уже внесенные данные, для получения данных из БД, для их сохранения, для обновления различных записей и для их удаления из БД. В число элементов DML-обращений входит основная часть SQL операторов.
  3. DCL (Data Control Language) - язык управления данными. Включает в себя запросы и команды, касающиеся разрешений, прав и других настроек СУБД.
  4. TCL (Transaction Control Language) - язык управления транзакциями. Конструкции такого типа применяют чтобы управлять изменениями, которые производятся с использованием DML запросов. Конструкции TCL позволяют нам производить объединение DML запросов в наборы транзакций.

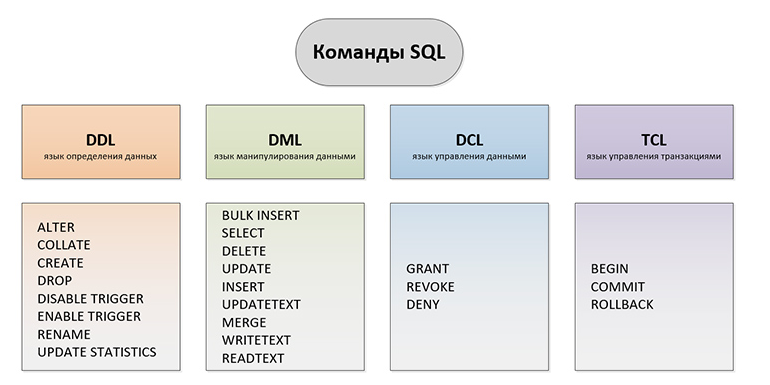


Рис. . Основные типы SQL запросов по их видам.

1. Схему базы данных в виде UML-диаграммы.

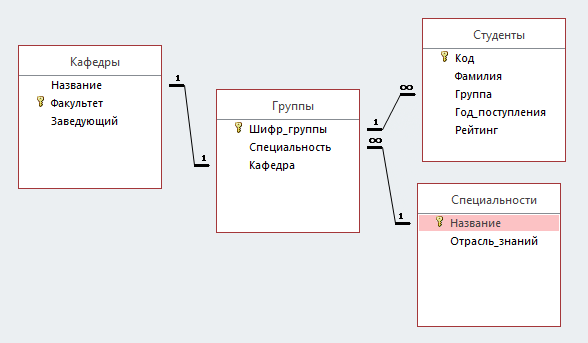


Рис. . Диаграмма связей между таблицами.

1. Скриншоты экрана приложения с результатом работы программы

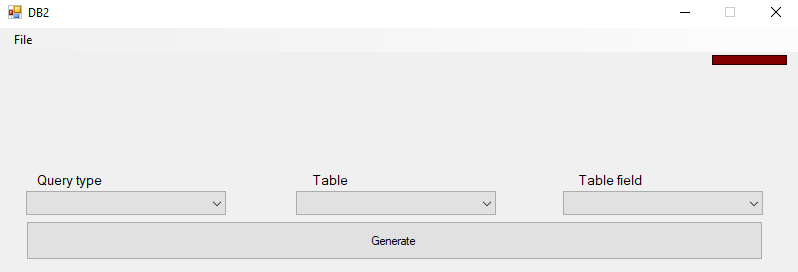


Рис. . Интерфейс программы.

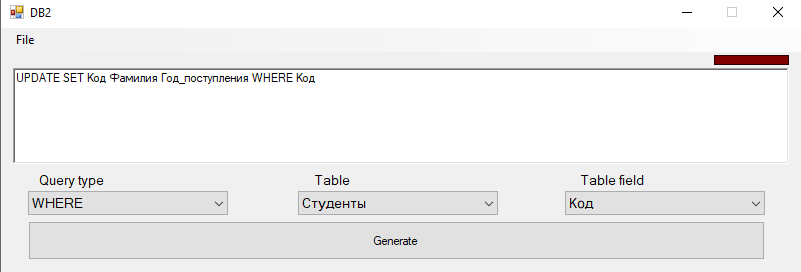


Рис. . Запрос на обновление полей таблицы в сыром виде.

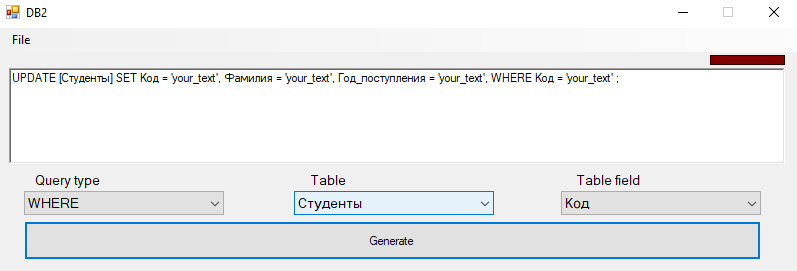


Рис. . Запрос на создание таблицы после нажатия кнопки Generate.

1. Текст исходных кодов программы

using System;

using System.Data;

using System.Windows.Forms;

using System.IO;

using System.Data.OleDb;

namespace DB2

{

public partial class Form1 : Form

{

private OleDbConnection connection = null;

public Form1()

{

InitializeComponent();

FormLoad();

}

private void ConnectDBClick(object sender, EventArgs e)

{

string connectionString = $"Provider = Microsoft.Jet.OLEDB.4.0; Data Source = ";

using (OpenFileDialog fd = new OpenFileDialog())

{

fd.Filter = "Access file(\*.mdb)|\*.mdb";

if (fd.ShowDialog() == DialogResult.Cancel)

return;

connectionString += fd.FileName + ";";

}

connection = new OleDbConnection(connectionString);

connection.Open();

}

private void CreateClick(object sender, EventArgs e)

{

richTextBox.Visible = true;

} private void OpenClick(object sender, EventArgs e)

{

using (StreamReader sr = new StreamReader("Query.txt"))

{

richTextBox.Text = sr.ReadToEnd();

}

}

private void SaveClicl(object sender, EventArgs e)

{

using (StreamWriter sw = new StreamWriter("Query.txt", false))

{

sw.WriteLine(richTextBox.Text);

}

}

private void GenerateBtnClick(object sender, EventArgs e)

{

GeneratorStage1();

GeneratorStage2();

int index = richTextBox.Text.Length - 2;

char c = richTextBox.Text[index];

if (richTextBox.Text[index] == ',')

{

richTextBox.Text = richTextBox.Text.Remove(index, 1);

richTextBox.Text += ";";

return;

}

if (richTextBox.Text[index] != ';')

richTextBox.Text += ";";

}

private void GeneratorStage1()

{

string[] keyWords = new string[] { "FROM", "UPDATE", "TABLE" };

string tempText = richTextBox.Text;

foreach (string s in keyWords)

{

int i = tempText.IndexOf(s, 0);

if (i == -1)

goto point1;

i += s.Length;

if (s == "TABLE")

{

tempText = tempText.Insert(i, " [Table\_name]");

goto point1;

}

tempText = tempText.Insert(i, $" [{comboBoxTable.Text}]");

point1:

richTextBox.Text = tempText;

}

}

private void GeneratorStage2()

{

string[] keyWords = new string[] { "SET", "WHERE" };

string tempText = richTextBox.Text;

foreach(string s in keyWords)

{

int i = tempText.IndexOf(s, 0);

if (i == -1)

goto point1;

i += s.Length;

int index = 0;

if (s == "SET")

{

index = tempText.IndexOf("WHERE", 0);

if (index != -1)

index = index - i;

}

string subStr = tempText.Substring(i, index);

string[] spliter = subStr.Split(' ');

subStr = null;

for (int j = 0; j < spliter.Length; j++)

{

if (spliter[j] != "")

tempText = tempText.Replace(spliter[j], spliter[j] + " = \'your\_text\',");

}

point1:

richTextBox.Text = tempText;

}

} // изм значения в полях таблицы

private void TableFieldSlctValChenged(object sender, EventArgs e)

{

if (richTextBox.Visible == false || comboBoxTableField.Text == "")

return;

if (comboBoxTableField.Text == "All")

{

richTextBox.Text += "\* ";

return;

}

richTextBox.Text += comboBoxTableField.Text + " ";

}

// изм значения в запросах

private void QueryTypeSlctValChenged(object sender, EventArgs e)

{

if (richTextBox.Visible == false || comboBoxQueryType.Text == "")

return;

richTextBox.Text += comboBoxQueryType.Text + " ";

}

// выпадение запросов

private void QueryTypeDrpDwn(object sender, EventArgs e)

{

try

{

comboBoxQueryType.Items.Clear();

using (StreamReader sr = new StreamReader("QueryTypes.txt"))

{

string val = null;

while ((val = sr.ReadLine()) != null)

comboBoxQueryType.Items.Add(val);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

// выпадение таблицы

private void TableDrpDwn(object sender, EventArgs e)

{

if (connection == null)

return;

comboBoxTable.Items.Clear();

DataTable tables = connection.GetSchema("Tables", new string[] { null, null, null, "TABLE" });

foreach (DataRow row in tables.Rows)

{

string tableName = row["TABLE\_NAME"].ToString();

comboBoxTable.Items.Add(tableName);

}

}

// выпадение полей таблицы

private void TableFieldDrpDwn(object sender, EventArgs e)

{

if (comboBoxTable.Text == "")

return;

comboBoxTableField.Items.Clear();

using (OleDbCommand command = new OleDbCommand($"SELECT \* FROM {comboBoxTable.Text} WHERE 1 = 0", connection))

{

OleDbDataReader reader = command.ExecuteReader();

DataTable table = reader.GetSchemaTable();

foreach (DataRow t in table.Rows)

{

string rowName = t.Field<string>("ColumnName");

comboBoxTableField.Items.Add(rowName);

}

comboBoxTableField.Items.Add("All");

reader.Close();

}

}

private void RedButtonClick(object sender, EventArgs e)

{

richTextBox.Clear();

}

private void ExitClick(object sender, EventArgs e)

{

Close();

}

private void FormClose(object sender, FormClosingEventArgs e)

{

connection.Close();

}

private void FormLoad()

{

richTextBox.Visible = false;

}

}

}

1. Краткие выводы

Изучил принципы физической организации баз данных. Изучил организацию схемы и структуры базы данных, структуры таблиц, типов данных в таблицах. Изучил основные типы запросов в базах данных – на создание БД, на создание таблиц, на выборку, на добавление, на удаление и на модификацию данных. Изучил интегрированные среды разработки ПО и программные средства управления базами данных

1. Перечень использованных источников
   1. <http://ru.it-dc.org/articles/umlfordevelopers?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>
   2. <https://www.microsoft.com/ru-ru/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling>
   3. <https://tproger.ru/translations/sql-recap/>
   4. <https://itvdn.com/ru/blog/article/m-sql>