Лабораторная работа по ADO.NET

**Цель лабораторной работы**: познакомиться с языком программирования С#, средствами разработки среды Visual Studio .Net 2003 и с использованием сервера баз данных SQL Server 2000, на примере создания клиентского Windows-приложения, работающего с БД.

**Задание:** написать приложение на Windows Forms позволяющее соединяться с удалённой базой данных, просматривать её содержимое, и выполнять стандартные SQL-запросы.

**Теория:**

SQL – это язык структурированных запросов для реляционных баз данных. SQL включает в себя следующие языковые подмножества:

• Data Definition Language (DDL) - определения данных;

• Data Manipulation Language (DML) - манипулирования данными;

• Transaction Control Language (TCL) - управление транзакциями;

• Data Control Language (DCL) - управление привилегий;

• Cursor Control Language (CCL) - управление курсором;

Язык DML содержит операторы, позволяющие выбирать, добавлять, удалять и модифицировать данные:

• SELECT – применяется для выборки данных;

• INSERT – применяется для добавления строк к таблице;

• DELETE – применяется для удаления строк из таблицы;

• UPDATE – применяется для исправления данных;

Рассмотрим более подробно использование оператора WHERE, который задаёт условие выборки данных. После оператора WHERE ( в команде SELECT) идёт выражение, задающее условие выборки строк из указанных таблиц базы. В выражении могут использоваться:

• операции отношения: = <> < > <= >=

• логические операции: OR, AND, NOT

• конструкции IS [ NOT ] NULL

• Может применяться вложенная выборка (SELECT);

**Код программы:**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data;

using System.Data.OleDb;

namespace PP\_LR6

{

public class DBRequest

{

private OleDbConnection DBCon = new OleDbConnection();

public void ConnectTo(string DataSource, string InitialCatalog)

{

DBCon.ConnectionString = "Provider=SQLOLEDB;" +

"Integrated Security=True;" +

"Data Source=" + DataSource + ";" +

"Initial Catalog=" + InitialCatalog + ";" +

"User ID=; Password=;";

try { DBCon.Open(); }

catch (Exception e) { throw e; }

}

public void Disconnect()

{

try

{

if (DBCon.State == ConnectionState.Open)

DBCon.Close();

}

catch

{

}

}

~DBRequest()

{

Disconnect();

}

public string GetTableFields(string TableName)

{

if (DBCon.State == ConnectionState.Open)

{

DataTable CurTable = new DataTable("ConnectedTab");

OleDbDataAdapter DBAdapt;

try

{

DBAdapt = new OleDbDataAdapter("SELECT \* FROM " + TableName, DBCon);

DBAdapt.Fill(CurTable);

}

catch (Exception e)

{

throw e;

}

string ResStr = "";

int ColCount = CurTable.Columns.Count;

for (int i = 0; i < ColCount; i++)

{

string StrCon = ", ";

if (i == ColCount - 1) StrCon = ";";

ResStr += CurTable.Columns[i].ColumnName + "[" + CurTable.Columns[i].DataType.Name + "]" + StrCon;

}

return ResStr;

}

else

return null;

}

public DataTable SQLRequest(string RequestStr)

{

if (DBCon.State == ConnectionState.Open)

{

DataTable ResultTab = new DataTable("SQLresult");

OleDbDataAdapter DBAdapt;

try

{

DBAdapt = new OleDbDataAdapter(RequestStr, DBCon);

DBAdapt.Fill(ResultTab);

}

catch (Exception e)

{

throw e;

}

return ResultTab;

}

else

return null;

}

}

}

///////////////////////////////////////////////////////////////////////////////////////////

using System;

using System.Data;

using System.Windows.Forms;

namespace PP\_LR6

{

public partial class MainForm : Form

{

// Класс выполняющий соединение с БД и запрос к ней:

public DBRequest UserReq;

// Таблица-результат выполнения запроса:

private DataTable RequestTab;

// Таблица структуры таблиц из БД:

private DataTable StructTab;

// Название Таблицы, обработанной последний раз в StructTab\_OnRowChanged

private string LastTabName;

public MainForm()

{

InitializeComponent();

}

private void MainForm\_Load(object sender, EventArgs e)

{

// Создаём класс взаимодействия с Базой Данных

UserReq = new DBRequest();

// Создаём таблицу и добавляем в неё столбцы:

StructTab = new DataTable("TabFields");

DataColumn NewDatCol = new DataColumn("Tables", Type.GetType("System.String"));

NewDatCol.AllowDBNull = false;

NewDatCol.Unique = true;

StructTab.Columns.Add(NewDatCol);

NewDatCol = new DataColumn("Fields", Type.GetType("System.String"));

NewDatCol.AllowDBNull = false;

NewDatCol.DefaultValue = "none;";

StructTab.Columns.Add(NewDatCol);

DatGridDBTables.DataSource = StructTab;

DatGridDBTables.ReadOnly = false;

datGridSQLResult.DataSource = RequestTab;

// Подключаем к таблице обработчик события изменения строки:

StructTab.RowChanged += new DataRowChangeEventHandler(StructTab\_OnRowChanged);

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Обработчик изменения строки в БД \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

private void StructTab\_OnRowChanged(object sender, DataRowChangeEventArgs e)

{

try

{

if (LastTabName != (string)e.Row["Tables"])

{

LastTabName = (string)e.Row["Tables"];

string Fields = UserReq.GetTableFields(LastTabName);

e.Row["Fields"] = Fields;

}

}

catch (Exception ex)

{

MessageBox.Show(this, ex.Message, "Connection Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\* Попытка Соединения с Базой Данных\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

private void btnConnect\_Click(object sender, System.EventArgs e)

{

bool Connected = false;

Cursor = Cursors.WaitCursor;

// Пробуем соединиться с БД:

try

{

UserReq.Disconnect();

UserReq.ConnectTo(tbDatSource.Text, tbInitCat.Text);

Connected = true;

}

catch (Exception e1)

{

MessageBox.Show(this, e1.Message, "Connection Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

Connected = false;

}

if (Connected)

{

// Открываем доступ к управлению выполнением запросов:

tbRequest.Enabled = true;

btnRequest.Enabled = true;

datGridSQLResult.Enabled = true;

DatGridDBTables.Enabled = true;

}

else

{

// Закрываем доступ к управлению выполнением запросов:

tbRequest.Enabled = false;

btnRequest.Enabled = false;

datGridSQLResult.Enabled = false;

DatGridDBTables.Enabled = false;

}

Cursor = Cursors.Arrow;

try

{

StructTab.Clear();

RequestTab.Clear();

}

catch

{

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Выполнение Заданного SQL-запроса \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

private void btnRequest\_Click(object sender, System.EventArgs e)

{

Cursor = Cursors.WaitCursor;

try

{

RequestTab = UserReq.SQLRequest(tbRequest.Text);

datGridSQLResult.DataSource = RequestTab;

}

catch (Exception ex)

{

MessageBox.Show(this, ex.Message, "Request Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

Cursor = Cursors.Arrow;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Выводим Данные Выбранной Таблицы \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

private void datGridDBTables\_MouseUp(object sender, System.Windows.Forms.MouseEventArgs e)

{

DataGridView.HitTestInfo HitInfo = DatGridDBTables.HitTest(e.X, e.Y);

if ((HitInfo.RowIndex >= 0) && (HitInfo.RowIndex < StructTab.Rows.Count))

{

tbRequest.Text = "SELECT \* FROM " + (string)StructTab.Rows[HitInfo.RowIndex]["Tables"];

btnRequest\_Click(this, null);

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Изменение Размеров Таблицы \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\*private void datGridDBTables\_Resize(object sender, System.EventArgs e)

{

int ColTablesWidth = DatGridDBTables.TableStyles[0].GridColumnStyles[0].Width;

DatGridDBTables.TableStyles[0].GridColumnStyles[1].Width =

DatGridDBTables.Width - ColTablesWidth - 57;

}\*/

}

}

**Скриншоты:**

После запуска кода представленный выше открывается форма, созданная с помощью конструктора. На данной форме представлены такие элементы, как Button-кнопка, одна для того чтобы подключится к базе данных, а другая для выполнения введенного запроса, также поля для ввода Sql запроса, имени сервера и каталога, а также DateGridView для вывода таблиц на форму.

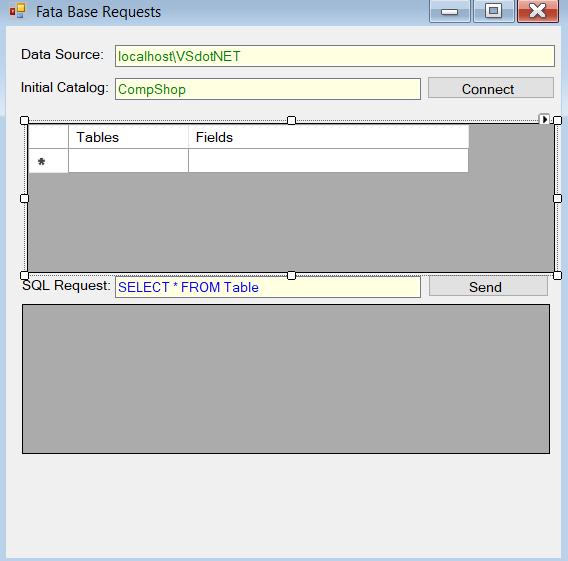


Рис.1-Форма при запуске.

При осуществлении запроса на вывод название компании, название материнской платы и адрес, а также при подключении через сервер к базе данных будет результат представленной ниже

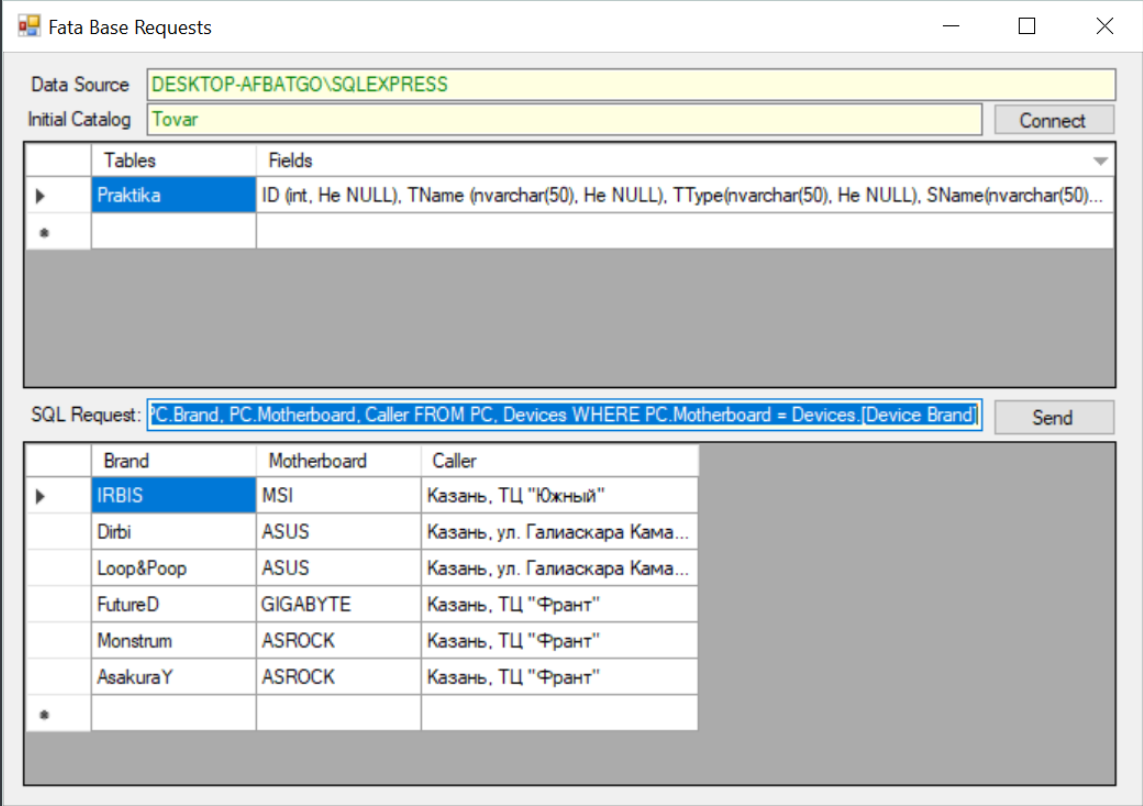


Рис.2-Форма после осуществления запроса.

Код SQL-запроса:

SELECT PC.Brand, PC.Motherboard, Caller

FROM PC, Devices

WHERE PC.Motherboard = Devices.[Device Brand]

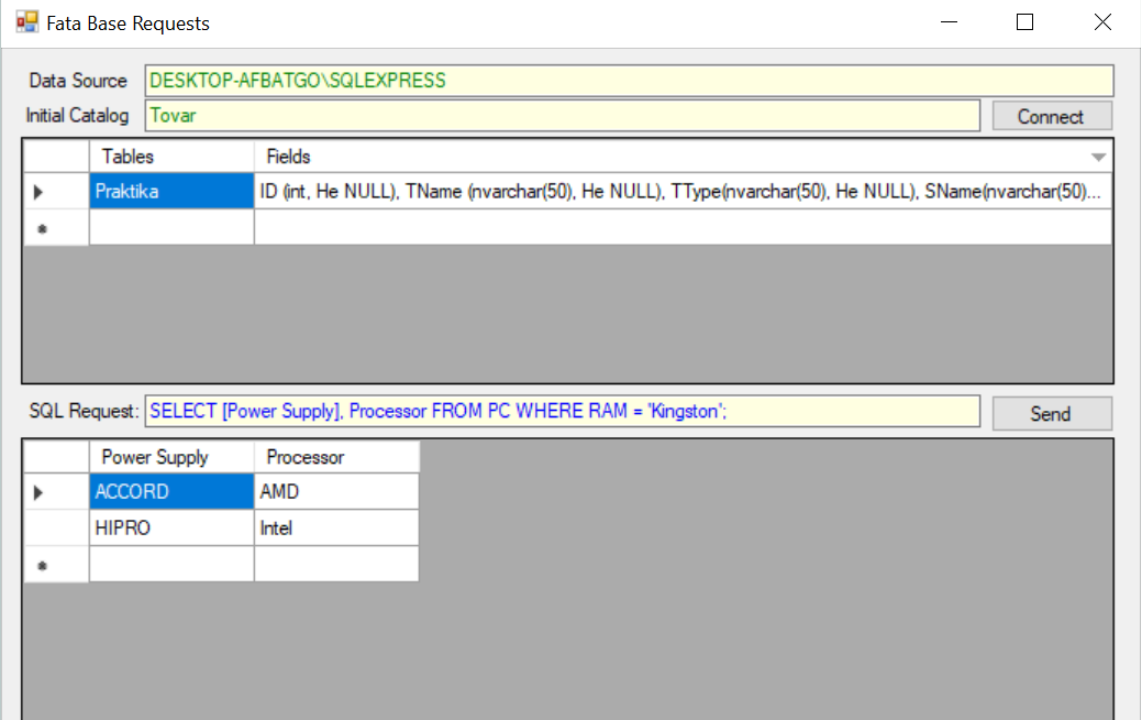


Рис.2-Результат запроса на вывод подходящих ОЗУ фирмы Кингстон производителей блоков питания и процессоров.

Код SQL-запроса:

SELECT [Power Supply], Processor

FROM PC

WHERE RAM = 'Kingston';

**Вывод:** познакомились с языком программирования С#, средствами разработки среды Visual Studio .Net 2015 и с использованием сервера баз данных SQL Server 2000, на примере создания клиентского Windows-приложения, работающего с БД.