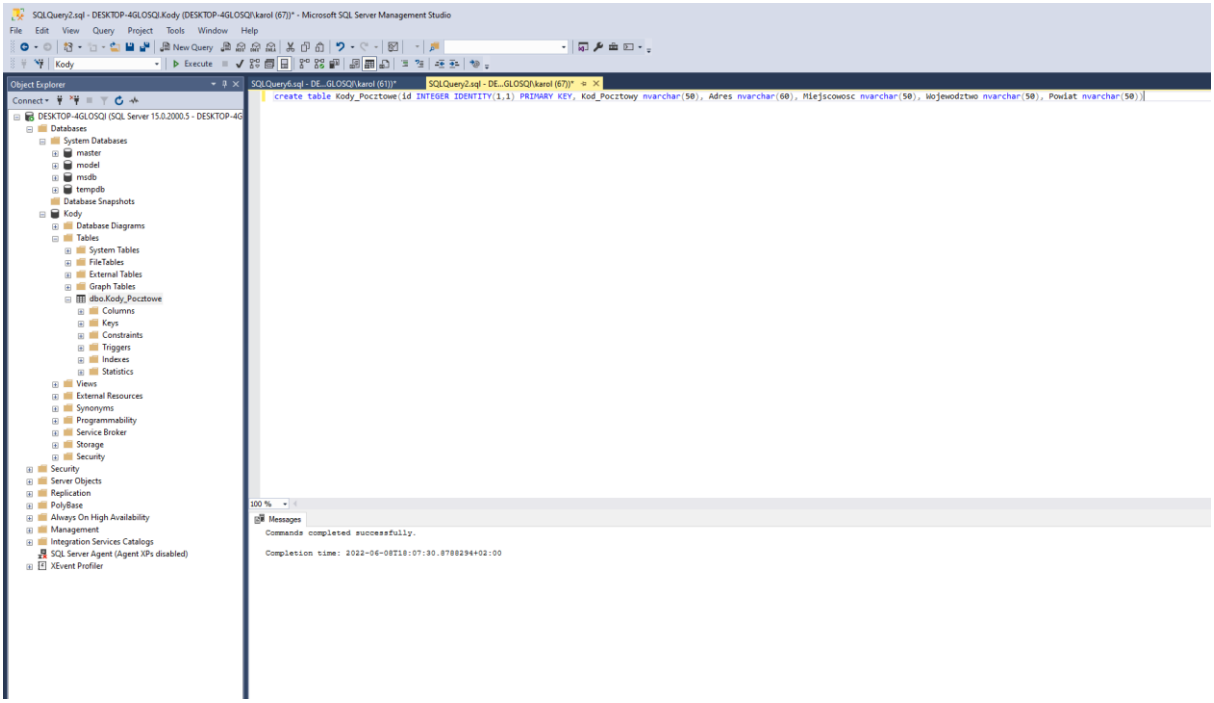
Zespół Szkół Elektrycznych w Gorzowie Wlkp.		
Informatyka		
Nr i temat ćwiczenia Zapoznanie się z podstawowymi zagadnieniami dotyczącymi łączenia się z bazą danych (Microsoft SQL Server) z poziomu aplikacji napisanej w C# (WinForms – Net.Framework, WPF, WinFormsc- .Net Core lub też winforms .Net 6) – wedle indywidualnych preferencji. Operacje Wykorzystanie dostawcy danych ADO.NET – SqlClient (**System.Data.SqlClient**) oraz ORM np. EF6. Główne zagadnienia realizowane w części teoretycznej ćwiczeń to: ➢ Omówienie podstawowych klas i metod niezbędnych do prawidłowego zainicjalizowania połączenia z bazą danych ➢ Ustanowienie połączenia ➢ Przechwytywanie błędów (**SqlException**) ➢ Wykonywanie podstawowych operacji z grupy **DQL** – Data Query Language. ➢ Zamykanie połączenia ➢ Wykorzystywanie bloku ***using.***	Nr w d 22,23	Klasa 3PT5
	Miesiąc Czerwiec	Data oddania sprawozdania 10.06.2022
Wykonał ćwiczenie: Karol Tokarek, Jakub Tokarek	Ocena pkt.	Podpis

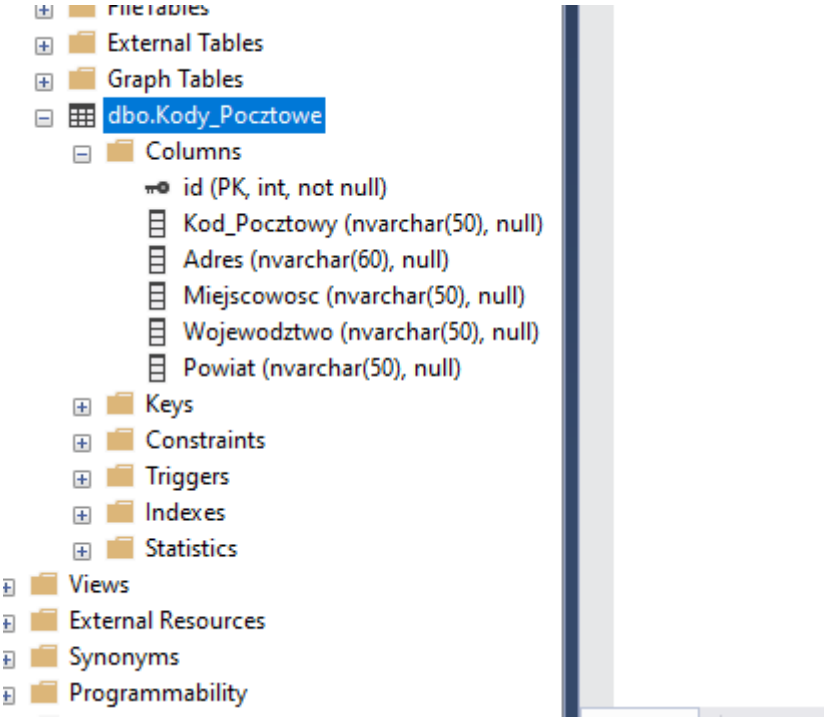
1.Cel ćwiczenia.

Zapoznanie się z podstawowymi zagadnieniami dotyczącymi łączenia się z bazą danych (Microsoft SQL Server) z poziomu aplikacji napisanej w C# (WinForms – Net.Framework, WPF, WinFormsc- .Net Core lub też winforms .Net 6) – wedle indywidualnych preferencji. Operacje Wykorzystanie dostawcy danych ADO.NET – SqlClient (**System.Data.SqlClient**) oraz ORM np. EF6. Poznam obsługę pliku .csv.

2.Opis / screeny:



Polecenie tworzące nową tabelę (później nvarchar został zamieniony na większą długość)



Stworzono poprawnie.

Bieżący wygląd aplikacji - po kliknięciu przycisku program odczytuje wybrany przez nas plik .csv, a następnie wybraną metodą (w kodzie) przesyła dane do bazy danych. Można też zobaczyć w tabeli podgląd tego pliku csv (mniej szary prostokąt).

Kod na następnej stronie.

KOD:

```
private void button1_Click(object sender, EventArgs e)
{
    sumka = 0;

        OpenFileDialog dialog = new OpenFileDialog();
        dialog.ShowDialog();
        int ImportedRecord = 0, inValidItem = 0;
        string SourceURl = "";
        txtFile.Text = dialog.FileName;


        DataTable dt = new DataTable();
    try
    {
        string[] lines = System.IO.File.ReadAllLines(dialog.FileName);
        button1.Text = "Wczytuje dane....";

        if (lines.Length > 0)
        {
            // addByWholeDataSqlBulkCopy(lines);
            //addByWholeDataSqlClient(lines);

            ///for wszystkie linie ///
            //TWORZENEI NAGLOWKA W TABELCE
            string firstLine = lines[0];
            string[] headerLabels = firstLine.Split(';');
            foreach (string headerWord in headerLabels)
            {
                dt.Columns.Add(new DataColumn(headerWord));
            }
            ///dla WYPELNIENIA DANYCH
            for (int i = 1; i < lines.Length; i++) //lines.Length
            {

                ////////// PO JEDNYM REKORDZIE /////////////////
                  addByOneByRecordbySqlClient(lines[i]);

                //addByOneByRecordbySqlBulkCopy(lines[i]);

                string[] dataWords = lines[i].Split(';');
                DataRow dr = dt.NewRow();
                int columnIndex = 0;
                foreach (string headerWord in headerLabels)
                {
                    dr[headerWord] = dataWords[columnIndex++];


                }
                dt.Rows.Add(dr);
            }
        }
        if (dt.Rows.Count > 0)
        {
            dataGridView1.DataSource = dt;

        }
        MessageBox.Show("Suma rekordu po rekordzie: " + sumka);
        MessageBox.Show("Srednia rekordu: " + (float)(sumka / lines.Length));
    }
    catch (Exception)
    {
        MessageBox.Show("error, zla sciezka lub plik / niepoprawny plik");
    }
    //addByWholeDataSqlClient(lines);
```

Metody zapisu:

- standardowo przy użyciu SqlClient (w tym przy użyciu klas SqlCommand, SqlConnection, SqlException)
    - (ss 1)”Metoda zapisu dotyczy pojedynczego rekordu (jako parametr metody przekazujemy jeden rekord) a wewnątrz metody otwierane jest połączenie do bazy danych, wykonywany jest zapis rekordu a następnie połączenie to jest zamykane. Tak więc następuje tyle wywołań metody ile jest rekordów w bazie danych.”
    - (ss 2) ”Metoda zapisu dotyczy całej kolekcji, a więc przekazywane są do niej wszystkie rekordy, wewnątrz metody na samym początku ustanawiane jest połączenie z bazą danych, następuje przesłanie wszystkich rekordów, a następnie rozłączenie i wyjście z metody.”

- przy użyciu SqlClient (w tym przy użyciu klas SqlBulkCopy)
    - (ss 3)”Metoda zapisu dotyczy pojedynczego rekordu (jako parametr metody przekazujemy jeden rekord) a wewnątrz metody otwierane jest połączenie do bazy danych, wykonywany jest zapis rekordu a następnie połączenie to jest zamykane. Tak więc następuje tyle wywołań metody ile jest rekordów w bazie danych.”
    - (ss 4) ”Metoda zapisu dotyczy całej kolekcji, a więc przekazywane są do niej wszystkie rekordy, wewnątrz metody na samym początku ustanawiane jest połączenie z bazą danych, następuje przesłanie wszystkich rekordów, a następnie rozłączenie i wyjście z metody.”

(1) Kod

```
private void addByOneByRecordbySqlClient(string one_record_in_csv)  ///dziala
{

    Stopwatch stopwatch = new Stopwatch();

//  MessageBox.Show("OD NOWA !!1");
//  for(int i=0; i<dataWords.Length; i++)
//  {
//      MessageBox.Show(dataWords[i]);
//  }

string sqlconn = "Server= localhost; Database = Kody; Integrated Security = SSPI";
    stopwatch.Start();

    // sqlconn = ConfigurationManager.ConnectionStrings["SqlCom"].ConnectionString;

    //MessageBox.Show("DO BAZY:" +one_record_in_csv); //insert into zapytanbie

    try
    {
        con = new SqlConnection(sqlconn);
        string[] dataWords = one_record_in_csv.Split(';');
        string sql = "INSERT INTO Kody_Pocztowe VALUES(@id, @username, @password, @email, @tel)";

        con.Open();
        comm = new SqlCommand(sql, con);
        comm.Parameters.AddWithValue("@id", dataWords[0]);
        comm.Parameters.AddWithValue("@username", dataWords[1]);
        comm.Parameters.AddWithValue("@password", dataWords[2]);
        comm.Parameters.AddWithValue("@email", dataWords[3]);
        comm.Parameters.AddWithValue("@tel", dataWords[4]);

        comm.ExecuteNonQuery();
        Console.WriteLine("Records Inserted Successfully");
        //  reader = comm.ExecuteReader();

        //  while (reader.Read())
        //  {
        // MessageBox.Show(reader.GetValue(0) + " - " + reader.GetValue(1) + " - " + reader.GetValue(2));
        // }
        //reader.Close();
       // comm.Dispose();

    }
    catch (SqlException ex)
    {
        MessageBox.Show("Can not  ! "+ex.ToString());
    }
    finally
    {
        //MessageBox.Show("koniec polaczenia");
        con.Close();
        stopwatch.Stop();
        sumka += stopwatch.Elapsed.TotalMilliseconds;
        //Debug.WriteLine(sumka);
        //MessageBox.Show("SUMA: " + stopwatch.Elapsed.TotalMilliseconds);
       // MessageBox.Show("TIME TO END; " + stopwatch.ElapsedMilliseconds.ToString());

    }


}
```

(2) Kod

```
private void addByWholeDataSqlClient(string[] wholedata)  ///dziala
{

    Stopwatch stopwatch = new Stopwatch();


    string sqlconn = "Server= localhost; Database= Kody; Integrated Security = SSPI";
    // sqlconn = ConfigurationManager.ConnectionStrings["SqlCom"].ConnectionString;
    stopwatch.Start();

    try
    {
        con = new SqlConnection(sqlconn);
       // string[] dataWords = one_record_in_csv.Split(';');
        con.Open();

        for (int i = 1; i < wholedata.Length; i++)
        {
            string wierszyk = wholedata[i];
            string[] dataWords = wierszyk.Split(';');

            string sql = "INSERT INTO Kody_Pocztowe VALUES(@id, @username, @password, @email, @tel)";
            comm = new SqlCommand(sql, con);
            comm.Parameters.AddWithValue("@id", dataWords[0]);
            comm.Parameters.AddWithValue("@username", dataWords[1]);
            comm.Parameters.AddWithValue("@password", dataWords[2]);
            comm.Parameters.AddWithValue("@email", dataWords[3]);
            comm.Parameters.AddWithValue("@tel", dataWords[4]);
            comm.ExecuteNonQuery();
            Console.WriteLine("Records Inserted Successfully");

          //  comm.Dispose();
        }

    }
    catch (SqlException ex)

    {
        MessageBox.Show("Can not  ! " + ex.ToString());
    }
    finally
    {
        //MessageBox.Show("koniec polaczenia");

        con.Close();
        stopwatch.Stop();
        MessageBox.Show("TIME TO END; " + stopwatch.ElapsedMilliseconds.ToString());
    }
}
```

(3) Kod

```
private void addByOneByRecordbySqlBulkCopy(string one_record_in_csv)
{
    Stopwatch stopwatch = new Stopwatch();
    string[] dataWords = one_record_in_csv.Split(';');
    stopwatch.Start();
    try
    {

        using (SqlConnection dbConnection = new SqlConnection("Server= localhost; Database = Kody; Integrated Sec
        {

            dbConnection.Open();

            //MessageBox.Show(dbConnection.State.ToString());
            using (SqlBulkCopy s = new SqlBulkCopy(dbConnection))
            {
                s.BatchSize = 10000;

                DataTable tbl = new DataTable();
                tbl.Columns.Add(new DataColumn("Kod_Pocztowy", typeof(string)));
                tbl.Columns.Add(new DataColumn("Adres", typeof(string)));
                tbl.Columns.Add(new DataColumn("Miejscowosc", typeof(string)));
                tbl.Columns.Add(new DataColumn("Wojewodztwo", typeof(string)));
                tbl.Columns.Add(new DataColumn("Powiat", typeof(string)));
                DataRow dr = tbl.NewRow();
                dr["Kod_Pocztowy"] = dataWords[0];
                dr["Adres"] = dataWords[1];
                dr["Miejscowosc"] = dataWords[2];
                dr["Wojewodztwo"] = dataWords[3];
                dr["Powiat"] = dataWords[4];
                tbl.Rows.Add(dr);

                s.DestinationTableName = "dbo.Kody_Pocztowe";
                s.ColumnMappings.Add("Kod_Pocztowy", "Kod_Pocztowy");
                s.ColumnMappings.Add("Adres", "Adres");
                s.ColumnMappings.Add("Miejscowosc", "Miejscowosc");
                s.ColumnMappings.Add("Wojewodztwo", "Wojewodztwo");
                s.ColumnMappings.Add("Powiat", "Powiat");
                s.WriteToServer(tbl);
                s.Close();

            }
            dbConnection.Close();
            stopwatch.Stop();
            sumka += stopwatch.Elapsed.TotalMilliseconds;

        }
    }
    catch(SqlException ex)
    {
        MessageBox.Show(ex.ToString());
    }
}
```

(4) Kod

```
Odwołania: 0
private void addByWholeDataSqlBulkCopy(string[] wholedata)
{

    Stopwatch stopwatch = new Stopwatch();
    stopwatch.Start();
    try {
        using (SqlConnection dbConnection = new SqlConnection("Server= localhost; Database = Kody; Integrated Securit
        {

            dbConnection.Open();

            //MessageBox.Show(dbConnection.State.ToString());
            using (SqlBulkCopy s = new SqlBulkCopy(dbConnection))
            {

                DataTable tbl = new DataTable();
                tbl.Columns.Add(new DataColumn("Kod_Pocztowy", typeof(string)));
                tbl.Columns.Add(new DataColumn("Adres", typeof(string)));
                tbl.Columns.Add(new DataColumn("Miejscowosc", typeof(string)));
                tbl.Columns.Add(new DataColumn("Wojewodztwo", typeof(string)));
                tbl.Columns.Add(new DataColumn("Powiat", typeof(string)));
                for (int i = 1; i < wholedata.Length; i++)
                {
                    string wierszyk = wholedata[i];
                    string[] dataWords = wierszyk.Split(';');

                    DataRow dr = tbl.NewRow();
                    dr["Kod_Pocztowy"] = dataWords[0];
                    dr["Adres"] = dataWords[1];
                    dr["Miejscowosc"] = dataWords[2];
                    dr["Wojewodztwo"] = dataWords[3];
                    dr["Powiat"] = dataWords[4];
                    tbl.Rows.Add(dr);


                }
                s.BatchSize = 10000;

                s.DestinationTableName = "dbo.Kody_Pocztowe";
                s.ColumnMappings.Add("Kod_Pocztowy", "Kod_Pocztowy");
                s.ColumnMappings.Add("Adres", "Adres");
                s.ColumnMappings.Add("Miejscowosc", "Miejscowosc");
                s.ColumnMappings.Add("Wojewodztwo", "Wojewodztwo");
                s.ColumnMappings.Add("Powiat", "Powiat");
                s.WriteToServer(tbl);
                s.Close();

            }
            dbConnection.Close();
            stopwatch.Stop();
            MessageBox.Show("TIME TO END; " + stopwatch.ElapsedMilliseconds.ToString());
        }




    }
    catch(SqlException ex)
    {
        MessageBox.Show("ERROR !" + ex.ToString());
    }
}
```

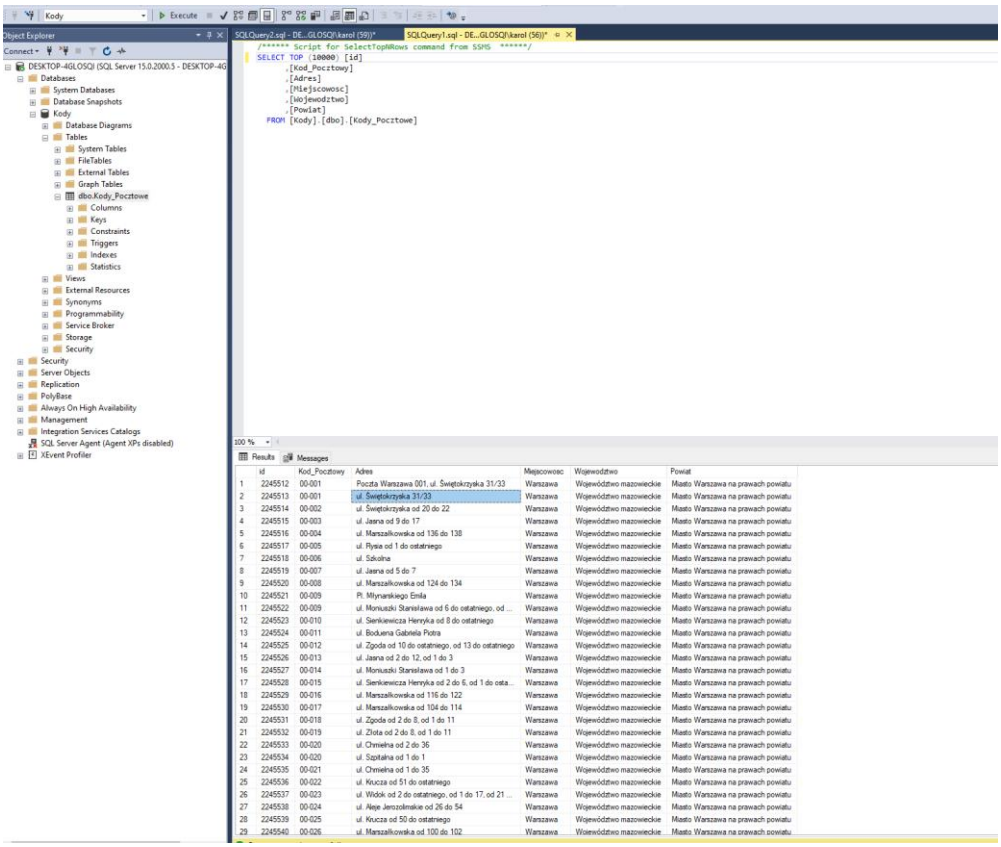
Tabela porównawcza wydajności poszczególnych metod i wnioski.

METODA	CZAS DLA WSZYSTKICH REKORDÓW	CZAS DLA POJEDYNCZEGO REKORDU
SqlClient (zwykłe) - rekord po rekordzie	~8800ms	~0,2ms
SqlClient (zwykłe) - wszystko w jednym poł.	~7500ms	~0.17ms
SqlClient (SqlBulkCopy) - rekord po rekordzie	~60s (60000ms)	~1,30ms
SqlClient (SqlBulkCopy) - wszystko w jednym poł.	~700ms	~0,01ms

Wnioski (na bazie tabeli):

Jak widzimy najwydajniejsza jest metoda, gdzie otwieramy jedno połączenie dla wielu rekordów. Otwieranie połączenie co rekord jest po prostu mało zoptymalizowane i kosztowne w czas i zasoby komputera. Najszybszą i najefektywniejszą metodą okazało się użycie klasy **SqlBulkCopy** – w niecałą sekundę wgrywa wszystkie ponad 40 tysięcy rekordów.

Dowód na tabelę uzupełnioną danymi:



3. Wnioski (ogólne):

Znamy zagadnienia dotyczące łączenia się z bazą danych (Microsoft SQL Server) z poziomu aplikacji napisanej w C# (WinForms). Operujemy przy wykorzystanie dostawcy danych ADO.NET – SqlClient (**System.Data.SqlClient**) ~~oraz ORM np. EF6.(?)~~

Potrafimy też odczytywać plik CSV i wczytywać jego zawartość do bazy danych.

*Dodatkowo: potrafimy posługiwać się i zarządzać bazami, tworzyć je, dodawać tabele, wykonywać zapytania ITP. w Microsoft SQL Management Studio.