

Krzysztof Tatar
 Przemysław Kleszcz
 Grupa Laboratoryjna: 2
 Studia Niestacjonarne II Stopnia
 Zaawansowane Systemy Baz Danych

Tematem naszej sprawozdania jest porównanie relacyjnej bazy Microsoft SQL Server oraz dokumentowej - CouchDB. Na podstawie wybranych baz danych postaramy się ukazać różnice pomiędzy relacyjnym a dokumentowym podejściem składowania danych.

Na samym początku w naszej prezentacji przedstawiliśmy podstawowe informacje na temat wybranych przez nas baz danych. Opracowaliśmy, krótkie zestawienie porównawcze obu systemów. Zaprezentowaliśmy wybrany wycinek rzeczywistości w postaci kilku tabel oraz połączeń pomiędzy nimi. Opisaliśmy w jaki sposób udało nam się dokonać migracji danych pomiędzy systemem relacyjnym a dokumentowym. Na samym końcu omówiliśmy sposób aktualizacji i pozyskiwania danych w obu tych systemach.

Jako przykład bazy danych wybraliśmy bazę Microsoft Adventure Works. Poniższy zrzut ekranu ukazuje naszą tabelę SQL Server wypełnioną danymi.

	BusinessEntityID	PersonType	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	EmailPromotion	AdditionalContactInfo
1	1	EM	0	NULL	Ken	J	Sánchez	NULL	0	NULL
2	2	EM	0	NULL	Teri	Lee	Duffy	NULL	1	NULL
3	3	EM	0	NULL	Roberto	NULL	Tamburello	NULL	0	NULL
4	4	EM	0	NULL	Rob	NULL	Walters	NULL	0	NULL
5	5	EM	0	Ms.	Gail	A	Erickson	NULL	0	NULL
6	6	EM	0	Mr.	Jossef	H	Goldberg	NULL	0	NULL
7	7	EM	0	NULL	Dylan	A	Miller	NULL	2	NULL
8	8	EM	0	NULL	Diane	L	Margheim	NULL	0	NULL
9	9	EM	0	NULL	Gigi	N	Matthew	NULL	0	NULL
10	10	EM	0	NULL	Michael	NULL	Raheem	NULL	2	NULL
11	11	EM	0	NULL	Ovidiu	V	Cracium	NULL	0	NULL
12	12	EM	0	NULL	Thierry	B	D'Hers	NULL	2	NULL
13	13	EM	0	Ms.	Janice	M	Galvin	NULL	2	NULL
14	14	EM	0	NULL	Michael	I	Sullivan	NULL	2	NULL
15	15	EM	0	NULL	Sharon	B	Salavaria	NULL	2	NULL
16	16	EM	0	NULL	David	M	Bradley	NULL	1	NULL
17	17	EM	0	NULL	Kevin	F	Brown	NULL	2	NULL
18	18	EM	0	NULL	John	L	Wood	NULL	2	NULL

Dane z Microsoft SQL Server zostały przez nas zmigrowane do CouchDB przy użyciu oprogramowania Talend Open Studio for Big Data. Poniższy zrzut ekranu ukazuje dane po stronie CouchDB w formacie JSON.

id "1"

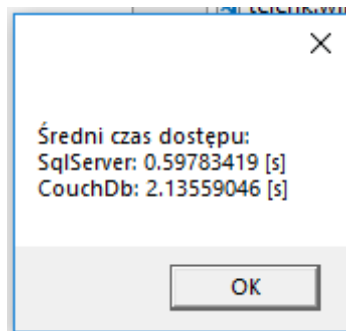
```
{
  "id": "1",
  "key": "1",
  "value": {
    "rev": "1-a4dd498d4a16a69e506aaae37e896977"
  },
  "doc": {
    "_id": "1",
    "_rev": "1-a4dd498d4a16a69e506aaae37e896977",
    "BusinessEntityID": "1",
    "PersonType": "EM",
    "NameStyle": "false",
    "Title": [],
    "FirstName": "Ken",
    "MiddleName": "J",
    "LastName": "Sánchez",
    "Suffix": [],
    "EmailPromotion": "0",
    "AdditionalContactInfo": [],
    "Demographics": "<IndividualSurvey xmlns=\"http://schemas.microsoft.com/sqlserver",
    "rowguid": "92C4279F-1207-48A3-8448-4636514EB7E2",
    "ModifiedDate": []
  }
}
```

id "10"

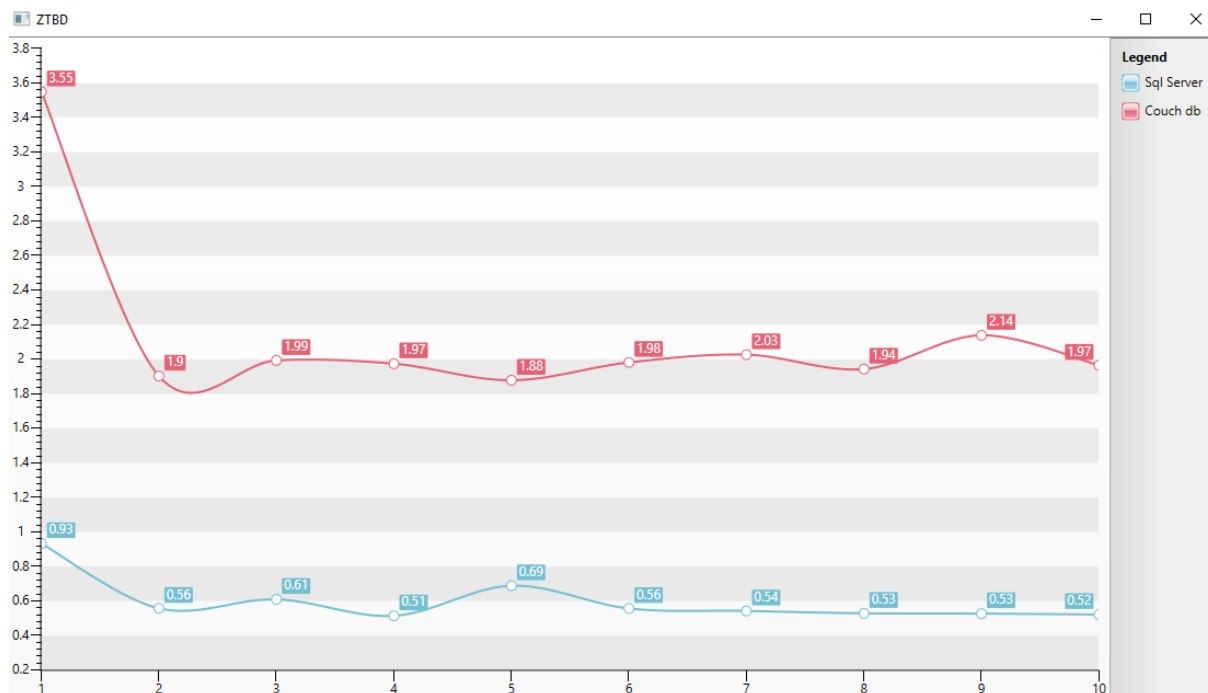
```
{
  "id": "10",
  "key": "10",
  "value": {
    "rev": "1-5ad3ffd4cc62df4a03040e386ea296e3"
  },
  "doc": {
    "_id": "10",
    "_rev": "1-5ad3ffd4cc62df4a03040e386ea296e3",

```

Po przeprowadzonej poprawnie migracji opracowaliśmy aplikację, która mierzy czas dostępu do bazy danych na podstawie prostego zapytania SQL (SQL Server) oraz analogicznego żądania http do serwera CouchDB pobierającego te same dane. Aplikacja po skończonej pracy wyświetla monit o czasie dostępu do poszczególnych baz danych:



W następnej kolejności program prezentuje wykres czasu dostępu poszczególnych prób pobrania danych z obu systemów bazodanowych. Można go zaobserwować poniżej:



Kod aplikacji:

```
namespace ZTBD
{
    public partial class MainWindow : Window
    {
        private List<PerformanceResult> CalculateSqlServer()
        {
            var results = new List<PerformanceResult>();
            var sw = new Stopwatch();
            var conString = Properties.Settings.Default.SqlServerConnectionString;
            for (var i = 1; i < Properties.Settings.Default.AttemptsCounter + 1; i++)
            {
                sw.Start();
                var data = new DataSet();
                var connection = new SqlConnection(conString);
                try
                {
                    connection.Open();
                    var sda = new SqlDataAdapter(Properties.Settings.Default.SqlServerQuery,
connection);
                    sda.Fill(data);
                    sw.Stop();
                    results.Add(new PerformanceResult(i, sw.Elapsed.TotalSeconds));
                }
                catch (Exception)
                {
                    MessageBox.Show("Wystąpił błąd.");
                    break;
                }
                finally
                {
                    connection.Close();
                }
                sw.Reset();
            }
            return results;
        }
        private List<PerformanceResult> CalculateCouchDB()
        {
            var results = new List<PerformanceResult>();
            var sw = new Stopwatch();
            var server = Properties.Settings.Default.CouchDbServer;
            var databaseName = Properties.Settings.Default.CouchDbDataBase;
            var sharpCouchDb = new DB();
            for (var i = 1; i < Properties.Settings.Default.AttemptsCounter + 1; i++)
            {
                sw.Start();
                sharpCouchDb.GetAllDocuments(server, databaseName);
                sw.Stop();
            }
        }
    }
}
```

```

        results.Add(new PerformanceResult(i, sw.Elapsed.TotalSeconds));
        sw.Reset();
    }
    return results;
}
private void ConfigureMapping(string label, List<PerformanceResult> results)
{
    var mapping = new SeriesMapping
    {
        LegendLabel = label,
        SeriesDefinition = new SplineSeriesDefinition(),
        ItemsSource = results
    };
    mapping.ItemMappings.Add(new ItemMapping("Attempt",
DataPointMember.XValue));
    mapping.ItemMappings.Add(new ItemMapping("Time", DataPointMember.YValue));
    radChart.SeriesMappings.Add(mapping);
}
public MainWindow()
{
    InitializeComponent();
    Loaded += MainWindow_Loaded;
}
private void MainWindow_Loaded(object sender, RoutedEventArgs e)
{
    var sqlServerResults = CalculateSqlServer();
    var couchDbResults = CalculateCouchDB();
    ConfigureMapping("Sql Server", sqlServerResults);
    ConfigureMapping("Couch db", couchDbResults);
    var sqlServerAvg = sqlServerResults.Select(x => x.Time).Average();
    var couchDbAvg = couchDbResults.Select(x => x.Time).Average();
    MessageBox.Show($"Średni czas
dostępu:\nSqlServer: {sqlServerAvg} [s]\nCouchDb: {couchDbAvg} [s]");
}
}
}

```

Aplikacja została opracowana za pomocą .NET Framework w języku C# (Visual Studio). Do stworzenia graficznego interfejsu użytkownika została użyta biblioteka WPF.

Sposób pozyskiwania i aktualizacji danych w bazie couchDB różni się znacząco od podejścia w standardowych bazach relacyjnych. Nie wykorzystujemy języka SQL. W zamian uzyskujemy dostęp do danych za pomocą języka JavaScript lub żądań HTTP. Opracowana przez nas aplikacja obrazuje, że czas dostępu do systemu CouchDB jest znacznie większy w stosunku do SQL Server. Wiąże się to z wykorzystanymi mechanizmami wyższego poziomu – serializacja danych oraz sposób ich przechowywania oraz narzutem czasowym narzuconym przez system (HTTP).