Aplikacja przetwarzająca własne funkcje agregujące w kodzie zarządzalnym.

Krzysztof Grządziel, R4IS2

23 stycznia 2010

Streszczenie

W artykule znajduje się opis wykonanych kroków potrzebnych do utworzenia własnych funkcji agregujących[1] w kodzie zarządzanym w bazie danych MS SQL Server 2008. Funkcje agregujące napisane zostały w języku C#. Pokazane zostało tworzenie assembly za pomocą Microsoft Visual Studio 2008, import assembly do bazy danych oraz stworzenie agregatów korzystających z naszej klasy. Konfiguracja MS SQL Server 2008 pozwalająca na połącznie się z bazą danych aplikacjom zewnętrznym. Na koniec pokazane zostały zapytania T-SQL na przykładowych danych, które wykorzystują nasze funkcje agregujące oraz wykresy z przetworzonymi danymi.

Spis treści

1 Własne funkcje agregujące

1.1 Tworzenie funkcji agregujących przy pomocy Visual Studio 2008

1. Otwieramy *Microsoft Visual Studio 2008*
2. klikamy File → New → Project
3. następnie Other Languages →
4. Visual C# → Database → SQL Server Project, ustawiamy pole name np. na MyAggregates, OK
5. Ustawiamy referencje do naszej bazy danych, OK
6. PPM[2] na MyAggregates → **Add** → Aggregate...
7. Ustawiamy Aggregate, pole name jako nazwe naszej funkcji np. AVGGeo.cs, **Add**
8. implementujemy naszą funkcję agregującą

[1] UDA – User Defined Aggregates
[2] prawy przycisk myszy

9. po zaimplementowaniu, klikamy Bulid → Build MyAggregates, aby skompilować
10. następnie Bulid → Deploy MyAggregates, Visual Studio zarejestruje za nas Assembly oraz napisane przez nas funkcje agregujące, ale **UWAGA!** tylko te, które przyjmują tylko jeden parametr
11. jeżeli napisalśmy funkcję agregującą, która przyjmuje więcej parametrów niż jeden, musimy ją zarejestrować ręcznie używając T-SQLa.

Przygotowaną solucję i funkcje agregujące można znaleźć w .\examples\p15_agregaty.zip

1.2 Rejestrowanie assembly i agregatów w bazie przy pomocy T-SQLa

Skompilować assembly z kodu zródłowego można za pomocą komendy w linii poleceń, np.

```
csc /target:library C:\AVGGeo.cs
```

Assembly oraz funkcje agregujące dodajemy za pomocą T-SQLa w nastepujący sposób (dla kodu znajdującego sie w .\examples\p15_agregaty.zip)

```
exec sp_configure 'clr enabled', 1
GO
RECONFIGURE WITH OVERRIDE
GO

IF EXISTS (SELECT name FROM sysobjects WHERE name = 'AVGWeighted')
   DROP AGGREGATE AVGWeighted
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'AVGHarm')
   DROP AGGREGATE AVGHarm
IF EXISTS (SELECT name FROM sysobjects WHERE name = 'AVGGeo')
   DROP AGGREGATE AVGGeo
GO

IF EXISTS (SELECT name FROM sys.assemblies WHERE name = 'MyAggregates')
   DROP ASSEMBLY MyAggregates
GO

CREATE ASSEMBLY MyAggregates FROM 'C:\Users\Administrator\Documents\hello\
    SqlServerProject8\AVGGeo\obj\Debug\SqlClassLibrary.dll'
GO

CREATE AGGREGATE AVGGeo (@value float) RETURNS float
EXTERNAL NAME MyAggregates.AVGGeo
CREATE AGGREGATE AVGHarm (@value float) RETURNS float
EXTERNAL NAME MyAggregates.AVGHarm
CREATE AGGREGATE AVGWeighted (@value float, @weight float) RETURNS float
EXTERNAL NAME MyAggregates.AVGWeighted

GO
```

2 Dane

2.1 Schemat tabel

Listing 1: Tworzenie tabeli spolki2

```
CREATE TABLE [dbo].[spolki2](
        [id] [int] NOT NULL,
        [nazwa] [nvarchar](50) NOT NULL,
        [symbol] [nvarchar](10) NOT NULL,
 CONSTRAINT [PK_spolki2] PRIMARY KEY CLUSTERED
(
        [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

W tabeli spolki2 trzymamy nazwy spółek.

Listing 2: Tworzenie tabeli notowania2

```
CREATE TABLE [dbo].[notowania2](
        [id] [int] NOT NULL,
        [spolki_id] [int] NOT NULL,
        [data] [date] NOT NULL,
        [otw] [decimal](10, 3) NOT NULL,
```

```
6           [min] [decimal](10, 3) NOT NULL,
7           [max] [decimal](10, 3) NOT NULL,
8           [zamnk] [decimal](10, 3) NOT NULL,
9           [zmiana] [decimal](10, 3) NOT NULL,
10          [wolumen] [int] NOT NULL,
11   CONSTRAINT [PK_notowania2] PRIMARY KEY CLUSTERED
12   (
13          [id] ASC
14   )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
15   ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
16   ) ON [PRIMARY]
```

W tabeli notowania2 przechowujemy dane na temat notowań spółek, przygotowane tak jak w punkcie (2.2).

2.2 Pozyskanie danych

Dane pobieramy z serwisu http://www.money.pl/gielda/archiwum/spolki/ za pomocą skryptu grab.sh, znajduje się on w (.\example\grab.sh).

Użycie:

```
1   ./grab.sh <symbol spolki> <katalog, do ktorego zapisujemy pobrane dane>
```

Przykład:

```
1   ./grab.sh TPS tpsa
2   ./grab.sh KGH kghm
```

2.3 Importowanie danych do tabel

Dodanie danych do spolki2

```
1   INSERT INTO dbo.spolki2 values(1,'KGHM', 'KGH'), (2,'TPSA','TPS'),
2   (3,'INGBSK', 'ING');
```

Aby zaimportować pozyskane dane do notowania2, dane muszą być odpowiednio przygotowane. W naszym przypadku oczyszczamy pobrane dane ze zbędnych linii, poleceniem (dla notowań TPSA)

```
1   sed -n '/^"200/p' ./tpsa/TPS-200* > tpsa-TPS.csv
```

dla pozostałych analogicznie. Następnie należy dodać na początku każdego wiersza, id rekordu (kolumna *id*), oraz id spółki, zgodny z numerem spółki w tabeli spolki2 (kolumna *spolki_id*) – np. importując dane do **Microsoft Office Excel**, dodając odpowiednie kolumny i eksportując dane spowrotem do pliku *.csv.

Tak przygotowany plik można znaleźć w .\examples\kghm_tpsa_ing.csv

Na koniec impotujemy plik z danymi do bazy danych

```
1   BULK INSERT dbo.notowania2
2   FROM 'C:\Users\Administrator\Desktop\kghm_tpsa_ing.csv'
3   WITH ( FIELDTERMINATOR = ';', ROWTERMINATOR = '\n' )
```

3 Zapytania SQL

Poniżej znajdują się przykładowe zapytania SQL dla danych z tabeli notowania2 z użyciem własnych funkcji agregujących.

3.1 Średnia krocząca

Zapytanie zwraca prostą średnią krocząca z 10 ostatnich próbek

```
1   SELECT x.data as xdata, AVG(y.wolumen)
2   FROM
3   (SELECT ROW_NUMBER() OVER (ORDER BY data ASC) AS Row, data FROM dbo.notowania2 where
        spolki_id = 2) AS x,
4   (SELECT ROW_NUMBER() OVER (ORDER BY data ASC) AS Row, data, wolumen FROM dbo.
        notowania2 where spolki_id = 2 ) AS y
5   WHERE x.Row between y.Row AND y.Row+9
6   GROUP BY x.data
7   ORDER BY x.data ASC
```

3.2 Średnia ważona

Zapytanie zwraca średnią ważoną kolumny zamnk z wagą zmiana grupując je miesiącami

```
1  SELECT CAST(data AS CHAR(7)) AS data, dbo.AVGWeighted(zamnk, zmiana) AS avgweight
2  FROM dbo.notowania2
3  WHERE spolki_id = 2 AND data BETWEEN '2009-01-01' AND '2009-12-31'
4  GROUP BY CAST(data as CHAR(7))
5  ORDER BY data ASC
```

3.3 Pozostałe średnie

Zapytanie zwracające prostą średnią kroczącą, kroczącą średnią geometryczną oraz kroczącą średnią harmoniczną z 10 ostatnich próblek dla akcji TPSA w przedziale od 2009-01-01 do 2009-12-31.

```
1  SELECT x.data AS xdata, AVG(y.wolumen) AS vol, dbo.AVGGeo(y.wolumen) AS avggeo, dbo.
       AVGHarm(y.wolumen) AS avgharm
2  FROM
3  (SELECT ROW_NUMBER() OVER (ORDER BY data ASC) AS Row, data FROM dbo.notowania2 WHERE
       spolki_id = 2) AS x,
4  (SELECT ROW_NUMBER() OVER (ORDER BY data ASC) AS Row, data, wolumen FROM dbo.
       notowania2 WHERE spolki_id = 2 ) AS y
5  WHERE x.data BETWEEN '2009-01-01' AND '2009-12-31' AND x.Row BETWEEN y.Row AND y.Row
       +9
6  GROUP BY x.data
7  ORDER BY x.data ASC
```

4 Konfiguracja MS SQL Server dla aplikacji PHP

Próba uruchomienia usługi Apache/2.2.14 (Win32) + PHP 5.3 (5.3.1) VC6 x86 Thread Safe dla Windowsa z włączonymi rozszerzeniami do obsługi połączeń z bazami danych MS SQL Server, kończy się niepowodzeniem. Z tego powodu należy pobrać rozszerzenia wspierane przez Microsoft ze strony SQL Server Driver for PHP 1.1 - October 2009 oraz zainstalować sterownik ODBC z Microsoft SQL Server Native Client.

4.1 Konfiguracja MS SQL Server

Aby zewnętrzne aplikacje miały dostęp do bazy danych, należy ustawić sposób autoryzacji na SQL Server Authentication.

1. Włączamy SQL Server Management Studio
2. otwieramy MSSQLSERVER Properties → Security
3. zmieniamy Server Authentication na SQL Server **and** Windows Authentication mode

Ustawiamy login i hasła do logowania się, używając Transact-SQLa

```
1  ALTER LOGIN sa ENABLE ; GO
2  ALTER LOGIN sa WITH PASSWORD = '<mocneHasloTutaj>' ; GO
```

lub przez Managment Studio

1. W Object Explorer, rozwijamy Security → Logins, PPM na sa i klikamy Properties
2. W General page ustawiamy hasło i potwierdzamy hasło dla loginu sa
3. W Status page, w sekcji Login ustawiamy Enabled i klikamy OK

Należy również zezwolić na połączeniea TCP/IP

1. Otwieramy Server Configuration Manager
2. SQL Server Network Configuration → Protocols for MSSQLSERVER
3. TCP/IP ustawaimy na Enable (ew. zmieniamy port we właściwościach)
4. następnie w drzewie wchodzimy do SQL Native Client 10.0 Configuration → Client Protocols i również TCP/IP ustawiamy na Enable
5. restartujemy serwer

Dodatkowo konfigurujemy firewall'a

1. Start menu → Ustawienia → Panel sterowania
2. Klikamy Centrum sieci i udostepniania i otwieramy Zapora sytemu Windows
3. następnie Zmien ustawienia → Wyjtki → Dodaj port...
4. nazwa SQL Serwer, port 1433, protokół TCP i OK

Teraz możemy przejść do uruchomienia aplikacji w PHP na zewnetrznym serwerze.

5 Aplikacja PHP

Napisana aplikacja w PHP pobiera dane z bazy MS SQL Server 2008 i generuje wykresy z pomocą biblioteki PHPlot. Aplikację można znaleźć w .\examples\phpapp\ .

5.1 Zapytania SQL wykorzystane w aplikacji

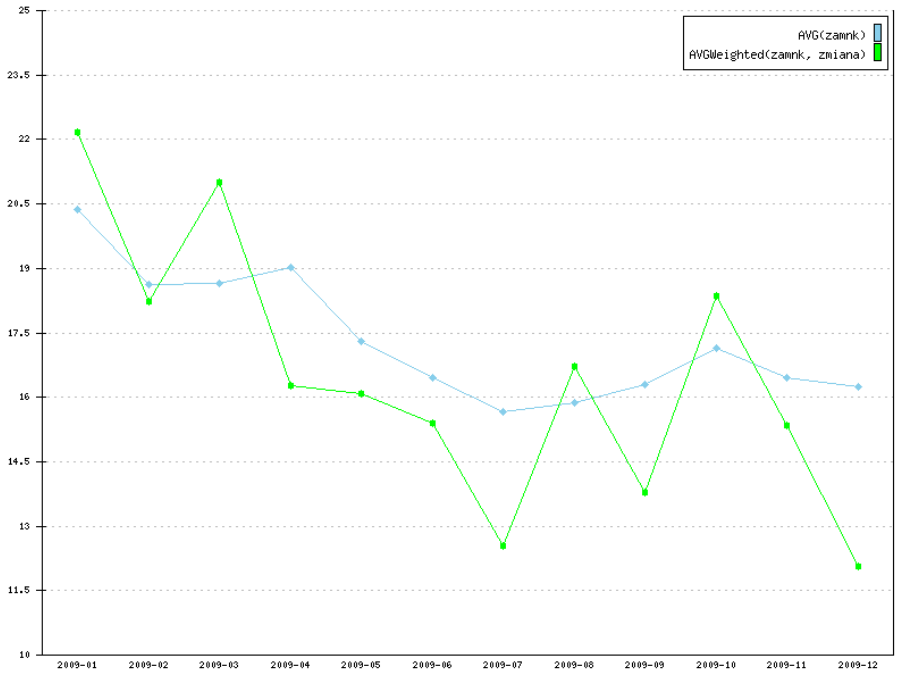
Zapytanie SQL pobierające dane do wygenerowania wykresu tpsa_weighted.png(1)

```
SELECT CAST(data AS CHAR(7)) AS data, avg(zamnk) AS avgzamnk, dbo.AVGWeighted(zamnk,
    zmiana) AS avgweight
FROM dbo.notowania2
WHERE spolki_id = 2 AND data BETWEEN '2009-01-01' AND '2009-12-31'
GROUP BY CAST(data AS CHAR(7))
ORDER BY data ASC
```

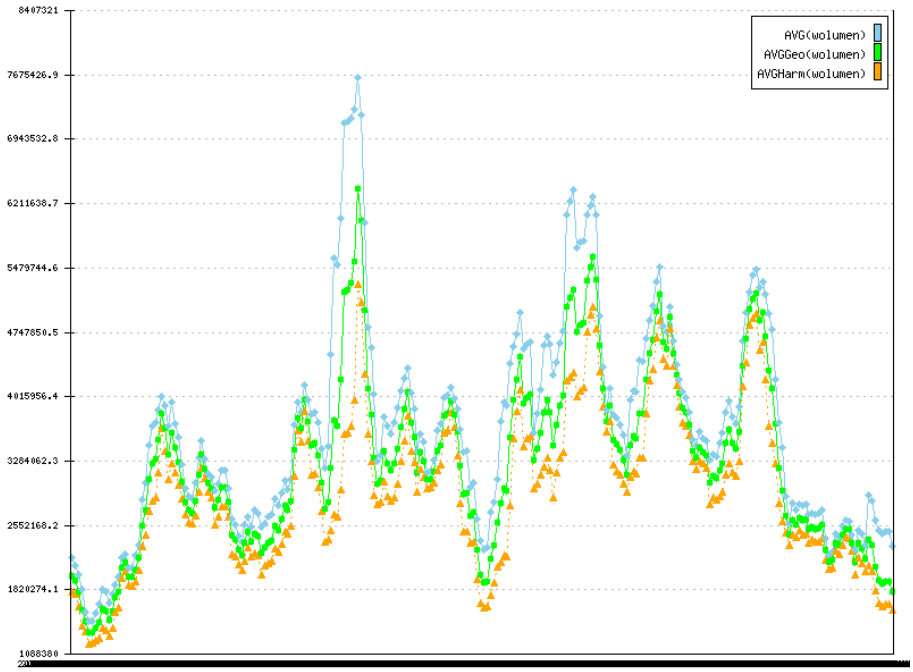
Zapytanie SQL pobierające dane do wygenerowania wykresu tpsa_kroczaca_avg_geo_harm.png(2)

```
SELECT x.data AS xdata, AVG(y.wolumen) AS vol, dbo.AVGGeo(y.wolumen) AS avggeo, dbo.
    AVGHarm(y.wolumen) AS avgharm
FROM
 (SELECT ROW_NUMBER() OVER (ORDER BY data ASC) AS Row, data FROM dbo.notowania2 WHERE
     spolki_id = 2) AS x,
 (SELECT ROW_NUMBER() OVER (ORDER BY data ASC) AS Row, data, wolumen FROM dbo.
    notowania2 WHERE spolki_id = 2 ) AS y
WHERE x.data BETWEEN '2009-01-01' AND '2009-12-31' AND x.Row between y.Row AND y.Row
    +9
GROUP BY x.data
ORDER BY x.data ASC
```

5.2 Wykresy wygenerowane przez aplikację



Rysunek 1: Średnia kolumny zamnk oraz średnia ważona kolumny zamnk z wagą zmiana pogrupowane miesiącami, dla akcji TPSA w 2009 roku.



Rysunek 2: Średnie kroczące: prosta, geometryczna, harmoniczna z ostatnich 10 próbek, dla akcji TPSA w 2009 roku.