

Importowanie danych do tabel SQL SERVER-a z wykorzystaniem polecenia BULK INSERT (ładowanie masowe)

Opracował: dr hab. inż. Artur Gramacki

1. Dane wejściowe(plik *MyCustomer.txt*)

- pola oddzielone tabulatorami,
- wartości tekstowe ujęte w cudzysłowy,
- w pierwszym wierszu nazwy kolumn (w cudzysłowach),
- data w formacie DD-MM-YYYY,
- separator dziesiętny w postaci kropki.

"CustomerKey"	"FirstName"	"LastName"	"BirthDate"	"Salary"
11000	"Jon"	"Yang"	06-10-1971	9000.00
11001	"Eugene"	"Huang"	01-05-1976	6000.33
11002	"Ruben"	"Torres"	09-02-1971	6000.45
11003	"Christy"	"Zhu"	14-08-1973	7000.00
11004	"Elizabeth"	"Johnson"	05-08-1979	8000.99
11005	"Julio"	"Ruiz"	01-08-1976	7000.00
11006	"Janet"	"Alvarez"	02-12-1976	7000.10
11007	"Marco"	"Mehta"	06-11-1969	6000.00
11008	"Rob"	"Verhoff"	04-07-1975	6000.90
11009	"Shannon"	"Carlson"	29-09-1969	7000.00

2. Tworzymy w SQL SERVER docelową tabelę (plik *CREATE_MyCustomer.sql*).

```
USE [AdventureWorksDW2014]
GO

DROP TABLE [dbo].[MyCustomer]
GO

CREATE TABLE [dbo].[MyCustomer] (
    [ID] [int] NOT NULL,
    [FirstName] [varchar](20) NOT NULL,
    [LastName] [varchar](50) NOT NULL,
    [BirthDate] [date] NOT NULL,
    [Salary] [decimal](8, 2) NOT NULL,
    CONSTRAINT [PK_MyCustomer] PRIMARY KEY([ID])
)
GO
```

3. Za pomocą narzędzia *bcp* (jest w katalogu *c:\Program Files\Microsoft SQL Server\Client SDK\ODBC\110\Tools\Binn*) tworzymy tzw. pliki z formatem (ang. *format file*). Możemy wygenerować starszą wersję czysto tekstową (ang. *Non-XML Format File*) oraz w formacie nowszym xml-owym (ang. *XML Format File*). Plik ten opisuje strukturę tabeli docelowej i będzie pomocny przy importowaniu do niej danych z zewnętrznego pliku tekstowego.

```
bcp AdventureWorksDW2014.dbo.MyCustomer format nul -c -x -f
MyCustomer.xml -t \t -T
```

```
bcp AdventureWorksDW2014.dbo.MyCustomer format nul -c -f
MyCustomer.fmt -t \t -T
```

W wyniku otrzymujemy dwa pliki:

```
<?xml version="1.0"?>
<BCDFORMAT xmlns="http://schemas.microsoft.com/sqlserver/2004/bulkload/format"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RECORD>
    <FIELD ID="1" xsi:type="CharTerm" TERMINATOR="\t" MAX_LENGTH="12"/>
    <FIELD ID="2" xsi:type="CharTerm" TERMINATOR="\t" MAX_LENGTH="20" COLLATION="SQL_Latin1_General_CP1_CI_AS"/>
    <FIELD ID="3" xsi:type="CharTerm" TERMINATOR="\t" MAX_LENGTH="50" COLLATION="SQL_Latin1_General_CP1_CI_AS"/>
    <FIELD ID="4" xsi:type="CharTerm" TERMINATOR="\t" MAX_LENGTH="11"/>
    <FIELD ID="5" xsi:type="CharTerm" TERMINATOR="\r\n" MAX_LENGTH="41"/>
  </RECORD>
  <ROW>
    <COLUMN SOURCE="1" NAME="ID" xsi:type="SQLINT"/>
    <COLUMN SOURCE="2" NAME="FirstName" xsi:type="SQLVARYCHAR"/>
    <COLUMN SOURCE="3" NAME="LastName" xsi:type="SQLVARYCHAR"/>
    <COLUMN SOURCE="4" NAME="BirthDate" xsi:type="SQLDATE"/>
    <COLUMN SOURCE="5" NAME="Salary" xsi:type="SQLDECIMAL" PRECISION="8" SCALE="2"/>
  </ROW>
</BCDFORMAT>
```

oraz

```
12.0
5
1  SQLCHAR      0  12  "\t"      1  ID          ""
2  SQLCHAR      0  20  "\t"      2  FirstName   SQL_Latin1_General_CP1_CI_AS
3  SQLCHAR      0  50  "\t"      3  LastName    SQL_Latin1_General_CP1_CI_AS
4  SQLCHAR      0  11  "\t"      4  BirthDate   ""
5  SQLCHAR      0  41  "\r\n"    5  Salary      ""
```

UWAGA: Jeżeli w trakcie uruchamiania polecenia *BULK INSERT* (poniżej) pojawi się komunikat o niewłaściwej wersji, winę za to ponosi pierwsza linia (12; oznaczenie wersji serwera: tu 2014). Należy ją zmienić na właściwą, np. dla serwera w wersji 2016 należy wpisać 13. Można też próbować użyć przełącznika *-V* w wywołaniu programu *bcp*. Poniżej przepisano odpowiedni tekst z oryginalnej dokumentacji

<https://docs.microsoft.com/en-us/sql/tools/bcp-utility?view=sql-server-2017>

-V (80 | 90 | 100 | 110 | 120 | 130)

Performs the bulk-copy operation using data types from an earlier version of SQL Server. This option does not prompt for each field; it uses the default values.

80 = SQL Server 2000 (8.x)

90 = SQL Server 2005 (9.x)

100 = SQL Server 2008 and SQL Server 2008 R2

110 = SQL Server 2012 (11.x)

120 = SQL Server 2014 (12.x)

130 = SQL Server 2016 (13.x)

For example, to generate data for types not supported by SQL Server 2000 (8.x), but were introduced in later versions of SQL Server, use the *-V80* option.

4. Przygotowujemy polecenie ładowania masowego danych (*BULK INSERT*). Zanim je uruchomimy, wcześniej kasujemy dane z tabeli docelowej oraz ustawiamy właściwy domyślny format daty. Pierwszy wiersz zawiera nagłówek, więc nie próbujemy go importować (*FIRSTROW* = 2). Patrz plik (*BULK_INSERT_MyCustomer.sql*)

```
DELETE FROM [dbo].[MyCustomer]
GO

-- Ustawiamy domyślny format daty.
-- Domyślnie w SQL Server jest yyyy-mm-dd
SET DATEFORMAT dmy
GO
```

```

BULK INSERT [dbo].[MyCustomer] FROM 'MyCustomer.txt'
WITH (
    CODEPAGE = 'ACP',
    DATAFILETYPE = 'char',
    KEEPIDENTITY,
    TABLOCK,
    FIRSTROW = 2,
    -- FORMATFILE = 'MyCustomer.xml',
    FORMATFILE = 'MyCustomer.fmt',
    ERRORFILE = 'MyCustomer.err'
)
GO

SELECT * FROM [dbo].[MyCustomer]
GO

```

Okazuje się, że użycie wygenerowanego pliku z formatem nie do końca działa poprawnie, gdyż cudzysłowy ograniczające pola tekstowe stają się częścią danych i w efekcie w tabeli pojawiają się dane jak poniżej:

	ID	FirstName	LastName	BirthDate	Salary
1	11000	"Jon"	"Yang"	1971-10-06	9000.00
2	11001	"Eugene"	"Huang"	1976-05-01	6000.00
3	11002	"Ruben"	"Torres"	1971-02-09	6000.00
4	11003	"Christy"	"Zhu"	1973-08-14	7000.00
5	11004	"Elizabeth"	"Johnson"	1979-08-05	8000.00
6	11005	"Julio"	"Ruiz"	1976-08-01	7000.00
7	11006	"Janet"	"Alvarez"	1976-12-02	7000.00
8	11007	"Marco"	"Mehta"	1969-11-06	6000.00
9	11008	"Rob"	"Verhoff"	1975-07-04	6000.00
10	11009	"Shannon"	"Carlson"	1969-09-29	7000.00

Należy więc trochę zmodyfikować plik z formatem, aby wyglądał jak poniżej:

```

12.0
5
1  SQLCHAR  0  12  "\t\"  1  ID  ""
2  SQLCHAR  0  20  "\"\t\"  2  FirstName  SQL_Latin1_General_CP1_CI_AS
3  SQLCHAR  0  50  "\"\t\"  3  LastName  SQL_Latin1_General_CP1_CI_AS
4  SQLCHAR  0  11  "\t\"  4  BirthDate  ""
5  SQLCHAR  0  41  "\r\n\"  5  Salary  ""

```

Wprowadzono zmiany mające na celu prawidłowe obsłużenie separatorów pól tekstowych (potrafisz wytłumaczyć, dlaczego tak a nie inaczej zmieniono zawartość pliku?). W wyniku otrzymujemy poprawne dane w polach tekstowych (nie ma już apostrofów):

Results		Messages			
	ID	FirstName	LastName	BirthDate	Salary
1	11000	Jon	Yang	1971-10-06	9000.00
2	11001	Eugene	Huang	1976-05-01	6000.33
3	11002	Ruben	Torres	1971-02-09	6000.45
4	11003	Christy	Zhu	1973-08-14	7000.00
5	11004	Elizabeth	Johnson	1979-08-05	8000.99
6	11005	Julio	Ruiz	1976-08-01	7000.00
7	11006	Janet	Alvarez	1976-12-02	7000.10
8	11007	Marco	Mehta	1969-11-06	6000.00
9	11008	Rob	Verhoff	1975-07-04	6000.90
10	11009	Shannon	Carlson	1969-09-29	7000.00

UWAGA: analogiczna zmiana w pliku xml niestety nie działa (generowane są błędy, sprawdź to sam/sama). Temat do wyjaśnienia.

5. Uwagi:

- W pliku źródłowym separatorami dziesiętnymi musi być kropka. Gdy separatorem będzie przecinek (jak wg. zasad języka polskiego), to niestety pojawiają się błędy importu (sprawdź to). Okazuje się, że w zaproponowanym mechanizmie importowania danych nie ma możliwości zdefiniowania innego niż domyślny separatora dziesiętnego. Przygotowując plik wejściowy (u nas *MyCustomer.txt*) należy zadbać, aby separator dziesiętny był poprawny.
- W pliku źródłowym w pierwszym wierszu są nazwy kolumn. Muszą one być ujęte w cudzysłowy. Na próbę usuń je i sprawdź, jaki jest efekt finalny importu (policz ilość zaimportowanych wierszy).
- Użyte narzędzie *bcp* ma dużo więcej możliwości niż zaprezentowane w instrukcji. Szczegóły patrz dokumentacja (np. [https://msdn.microsoft.com/pl-pl/library/ms162802\(v=sql.110\).aspx](https://msdn.microsoft.com/pl-pl/library/ms162802(v=sql.110).aspx))
- Powodzenie importu danych zewnętrznych do tabel SQL SERVER-a zależy w zdecydowanym stopniu od poprawnego przygotowania plików z danymi źródłowymi. W instrukcji zwrócono tylko uwagę na trzy aspekty: format daty, ogranicznik pól tekstowych (cudzysłowy) oraz separator dziesiętny. W żadnym wypadku nie wyczerpuje to zagadnienia i w praktyce może pojawić się jeszcze bardzo wiele innych problemów.

6. Wykorzystanie narzędzia *bcp* do eksportowania danych. Wykonajmy w tej chwili następujące polecenie:

```
bcp AdventureWorksDW2014.dbo.MyCustomer out MyCustomer_exp.txt -T -c
-t \t -r \n
```

Starting copy...

10 rows copied.

Network packet size (bytes): 4096

Clock Time (ms.) Total : 15 Average : (666.67 rows per sec.)

którego efektem działania będzie powstanie pliku tekstowego z następującą zawartością:

```
11000      Jon      Yang      1971-10-06  9000.00
```

11001	Eugene	Huang	1976-05-01	6000.33
11002	Ruben	Torres	1971-02-09	6000.45
11003	Christy	Zhu	1973-08-14	7000.00
11004	Elizabeth	Johnson	1979-08-05	8000.99
11005	Julio	Ruiz	1976-08-01	7000.00
11006	Janet	Alvarez	1976-12-02	7000.10
11007	Marco	Mehta	1969-11-06	6000.00
11008	Rob	Verhoff	1975-07-04	6000.90
11009	Shannon	Carlson	1969-09-29	7000.00

7. Wykorzystanie narzędzia *bcp* do importowania danych. Tym razem dane będziemy importowali do tabeli nie z wykorzystaniem ładowania masowego danych BULK INSERT, a wykorzystamy zewnętrzny (w stosunku do SQLServer-a) program *bcp*. W pierwszej kolejności utworzymy nową tabelę *MyCustomer2*, o strukturze identycznej jak tabela *MyCustomer* (patrz punkt 2). Następnie wykonajmy następujące polecenie (pamiętajmy, że musimy utworzyć wcześniej plik z formatem, patrz 3, lub wykorzystać plik z wcześniejszych punktów, nieznacznie go tylko modyfikując ręcznie):

```
bcp AdventureWorksDW2014.dbo.MyCustomer in MyCustomer_exp.txt
-T -f \MyCustomer2.fmt
```

Starting copy...

10 rows copied.

Network packet size (bytes): 4096

Clock Time (ms.) Total : 16 Average : (625.00 rows per sec.)

W efekcie w tabeli pojawiają się zaimportowane dane:

Results		Messages			
	ID	FirstName	LastName	BirthDate	Salary
1	11000	Jon	Yang	1971-10-06	9000.00
2	11001	Eugene	Huang	1976-05-01	6000.33
3	11002	Ruben	Torres	1971-02-09	6000.45
4	11003	Christy	Zhu	1973-08-14	7000.00
5	11004	Elizabeth	Johnson	1979-08-05	8000.99
6	11005	Julio	Ruiz	1976-08-01	7000.00
7	11006	Janet	Alvarez	1976-12-02	7000.10
8	11007	Marco	Mehta	1969-11-06	6000.00
9	11008	Rob	Verhoff	1975-07-04	6000.90
10	11009	Shannon	Carlson	1969-09-29	7000.00

8. Podsumowując wykonaliśmy w ćwiczeniu następujące czynności:
- Utworzyliśmy tabelę *MyCustomer*,
 - Utworzyliśmy na bazie tej tabeli plik z formatem *MyCustomer.fmt*,
 - Za pomocą narzędzia *bcp* załadowaliśmy do tabeli *MyCustomer* dane z tabeli *MyCustomer.txt*,
 - Poprawiliśmy ręcznie zawartość pliku *MyCustomer.fmt*, aby pozbyć się niechcianych cudzysłówów w tabeli *MyCustomer*. Używając *bcp* ponownie załadowaliśmy dane do tej tabeli,
 - Za pomocą narzędzia *bcp* wyeksportowaliśmy zawartość tabeli *MyCustomer* do pliku *MyCustomer_exp.txt*,

- f) Utworzyliśmy nową tabelę *MyCustomer2*, o strukturze analogicznej to struktury tabeli *MyCustomer*,
- g) Utworzyliśmy nowy plik z formatem *MyCustomer2.fmt*, który następnie wykorzystaliśmy w programie *bcp* do wykonania importu danych z pliku *MyCustomer_exp.txt* do tabeli *MyCustomer2*.