



**Wydział
Elektryczny**

POLITECHNIKA WARSZAWSKA

1DI1532:A/Z5 - Usuwanie kolumny pomimo ograniczeń

Sprawozdanie

Opis problemu

Piątym zadaniem realizowanym w ramach laboratorium z Administrowania Bazami Danych jest umożliwienie usuwania kolumn w wybranej tabeli, w wybranej bazie, których usunięcie może być ograniczone przez istniejące *constrainty*. Przykład takiego ograniczenia dla prostej tabeli przykładowej pokazuje poniższy zrzut ekranu.

```

85 CREATE TABLE test_us_kol ([id] nchar(6), czy_wazny bit NOT NULL default 0)
86 go
87 /* próbuję usunąć czy_wazny z tabeli
88 */
89 ALTER TABLE test_us_kol drop column czy_wazny
90
91 SELECT * FROM test_us_kol
92 /***** Test procedury *****/
93 EXEC dbo.delete_column @db = 'z4_baza', @table = 'test_us_kol', @column = 'czy_wazny'
94 /*****

```

Messages

Msg 5074, Level 16, State 1, Line 89
The object 'DF_test_us_k_czy_w_5070F446' is dependent on column 'czy_wazny'.
Msg 4922, Level 16, State 9, Line 89
ALTER TABLE DROP COLUMN czy_wazny failed because one or more objects access this column.

Completion time: 2020-12-21T00:52:24.1492068+01:00

Rys.1 Brak możliwości usunięcia kolumny

Jak widać, poprzez ograniczenie na tabeli w postaci obiektu DF_test_us_k_czy_w_5070F446 nie jest możliwe usunięcie tej kolumny. Celem zadania jest umożliwienie takiego usuwania w postaci procedury, w której jako parametry podajemy nazwę bazy, tabeli oraz kolumny, którą chcemy usunąć.

Opis realizacji zadania

W zrealizowanej procedurze (kod pod koniec dokumentu) najpierw sprawdzam czy wybrana kolumna istnieje. Jeśli nie, zwracany jest odpowiedni błąd. Wygląda on następująco:

```

91 SELECT * FROM test_us_kol
92 /***** Test procedury *****/
93 EXEC dbo.delete_column @db = 'z4_baza', @table = 'test_us_kol', @column = 'bledna_kolumna';
94 /*****

```

Messages

(0 rows affected)
Msg 50000, Level 11, State 1, Procedure dbo.delete_column, Line 19 [Batch Start Line 92]
BŁĄD! NIE MA TAKIEJ KOLUMNY

Completion time: 2020-12-21T00:56:54.6508004+01:00

Rys.2 Wywołanie procedury dla nieistniejącej kolumny

Procedura sprawdza, jakie ograniczenia stoją na przeszkodzie usunięcia kolumny, a następnie, w pętli, usuwa wszystkie te ograniczenia, aby na koniec spokojnie usunąć kolumnę.

Poniżej zaprezentowałem działanie procedury. Zapewniłem w niej wrażliwość na powtarzające się nazwy kolumn w różnych tabelach -- kolumna zostanie usunięta tylko w tabeli podanej jako argument.

```

89
90 SELECT * FROM test_us_kol
91 /***** Test procedury *****/
92 EXEC dbo.delete_column @db = 'z4_baza', @table = 'test_us_kol', @column = 'czy_wazny';
93 /*****/

```

100 %

Results Messages

id	czy_wazny
----	-----------

Rys.3 Tabela przed wykonaniem procedury

```

90 SELECT * FROM test_us_kol
91 /***** Test procedury *****/
92 EXEC dbo.delete_column @db = 'z4_baza', @table = 'test_us_kol', @column = 'czy_wazny';
93 /*****/

```

100 %

Results Messages

constraint_name
DF_test_us_k_czy_w_5070F446

Rys.4 W procedurze wyświetliłem ograniczenia, które w tym przypadku są usuwane przed usunięciem kolumny

```

90 SELECT * FROM test_us_kol
91 /***** Test procedury *****/
92 EXEC dbo.delete_column @db = 'z4_baza', @table = 'test_us_kol', @column = 'czy_wazny';
93 /*****/

```

100 %

Results Messages

id

Rys.5 Tabela po wykonaniu procedury -- kolumna poprawnie usunięta, dzięki pozbyciu się ograniczeń wewnątrz procedury

Kod procedury

```

/***** Z5 Maciej Czarkowski 292810 *****/
USE z4_baza
GO

/***** Procedura do usuwania *****/
IF NOT EXISTS
(
    SELECT 1
        from sysobjects o (NOLOCK)
        WHERE (o.[name] = 'delete_column')
        AND (OBJECTPROPERTY(o.[ID], 'IsProcedure')=1)
)
BEGIN
    DECLARE @stmt nvarchar(100)
    SET @stmt = 'CREATE PROCEDURE dbo.delete_column AS '
    EXEC sp_sqlexec @stmt
END
GO

ALTER PROCEDURE dbo.delete_column (@db nvarchar(100), @table nvarchar(100), @column
nvarchar(100))
AS
BEGIN
    DECLARE @sql nvarchar(1000), @constraint nvarchar(256)
    CREATE TABLE #FK ([constraint_name] varchar(30))
    CREATE TABLE #CHECK ([column_name] varchar(30))

    -- Sprawdzam czy kolumna istnieje w danej tabeli
    SET @sql = N'USE [' + @db + N']; '
    + 'INSERT INTO #CHECK '
    + 'SELECT 1 FROM sys.columns sc JOIN sys.objects so ON sc.object_id=so.object_id'
    + ' WHERE sc.name=''' + @column + ''' AND so.name=''' + @table + ''''
    --SELECT @sql
    EXEC sp_sqlexec @sql
    IF NOT EXISTS
    (
        SELECT 1 FROM #CHECK
    )
    BEGIN
        RAISERROR('BŁĄD! NIE MA TAKIEJ KOLUMNY', 11, 1);
        RETURN -1
    END

    -- Sprawdzam jakie ograniczenia będą do zdjęcia
    SET @sql = N'USE [' + @db + N']; '
    + 'INSERT INTO #FK '
    + 'SELECT so.name FROM sys.objects so '
    + 'JOIN sys.columns sc ON so.parent_object_id=sc.object_id '
    + 'JOIN sys.tables st ON st.object_id=sc.object_id '
    + 'WHERE sc.name=''' + @column + ''' AND st.name=''' + @table + ''''

    --SELECT @sql
    EXEC sp_sqlexec @sql
    --SELECT * FROM #FK

```

```

DECLARE CC INSENSITIVE CURSOR FOR
    SELECT *
        FROM #FK f
        ORDER BY 1

OPEN CC
FETCH NEXT FROM CC INTO @constraint

-- W pętli zdejmuję ograniczenia
WHILE (@@FETCH_STATUS = 0)
BEGIN
    SET @sql = 'USE [' + @db + N']; '
    + N'ALTER TABLE ' + @table + N' DROP CONSTRAINT ' + @constraint
    EXEC sp_sqlexec @sql
    FETCH NEXT FROM CC INTO @constraint
END
CLOSE CC
DEALLOCATE CC

-- Po usunięciu ograniczeń usuwam kolumnę
SET @sql = 'USE [' + @db + N']; '
    + N'ALTER TABLE ' + @table + N' DROP COLUMN ' + @column
EXEC sp_sqlexec @sql
END
GO

/***** Zapytania testowe *****/
SELECT * FROM sys.columns

DROP TABLE test_us_kol

CREATE TABLE test_us_kol ([id] nchar(6), czy_wazny bit NOT NULL default 0)
go
/* próbuję usunąć czy_wazny z tabeli
*/
ALTER TABLE test_us_kol drop column czy_wazny

SELECT * FROM test_us_kol
/***** Test procedury *****/
EXEC dbo.delete_column @db = 'z4_baza', @table = 'test_us_kol', @column = 'czy_wazny';
/*****

```