Błędy i Transakcje

Zastosowanie ustawienia błędu do wyświetlenia komunikatu

```
RAISERROR ('Komunikat',
1,                  -- grupa błędów.
1);                 -- informacja dodatkowa  0÷255.
```

Komunikat

Msg 50000, Level 1, State 1

```
RAISERROR ('Komunikat',
10,                 -- grupa błędów.
1);          -- informacja dodatkowa  0÷255.
```

Komunikat

```
RAISERROR ('Komunikat',
11,                 -- grupa błędów.
1);          -- informacja dodatkowa  0÷255
```

Msg 50000, Level 11, State 1, Line 1

Komunikat

0-18 zakres dostępny dla każdego użytkownika
19-25 tylko dla sysadmina lub użytkownika z uprawnieniami ALTER TRACE przy użyciu opcji WITH LOG (bledy od 20 do 25 traktowane są jako krytyczne)

Msg 2754, Level 16, State 1, Line 1

Error severity levels greater than 18 can only be specified by members of the sysadmin role, using the WITH LOG option.

Zastosowanie ustawienia błędu do wyświetlenia komunikatu

```
RAISERROR ('Komunikat',
19,                          -- grupa błędów.
1)WITH LOG
```

```
Msg 50000, Level 19, State 1, Line 1
Komunikat
```

```
RAISERROR ('Komunikat',
20,                -- grupa błędów.
1)WITH LOG
```

```
Msg 2745, Level 16, State 2, Line 1
Process ID 54 has raised user error 50000, severity 20. SQL Server is terminating this process.
Msg 50000, Level 20, State 1, Line 1
Komunikat
Msg 0, Level 20, State 0, Line 0
W bieżącym poleceniu wystąpił poważny błąd. Ewentualne wyniki powinny zostać odrzucone.
```

Jeśli ustawimy poziom większy niż 25 i tak zostanie potraktowany jako 25

0-18 zakres dostępny dla każdego użytkownika
19-25 tylko dla sysadmina lub użytkownika z uprawnieniami ALTER TRACE przy użyciu opcji WITH LOG (bledy od 20 do 25 traktowane są jako krytyczne)

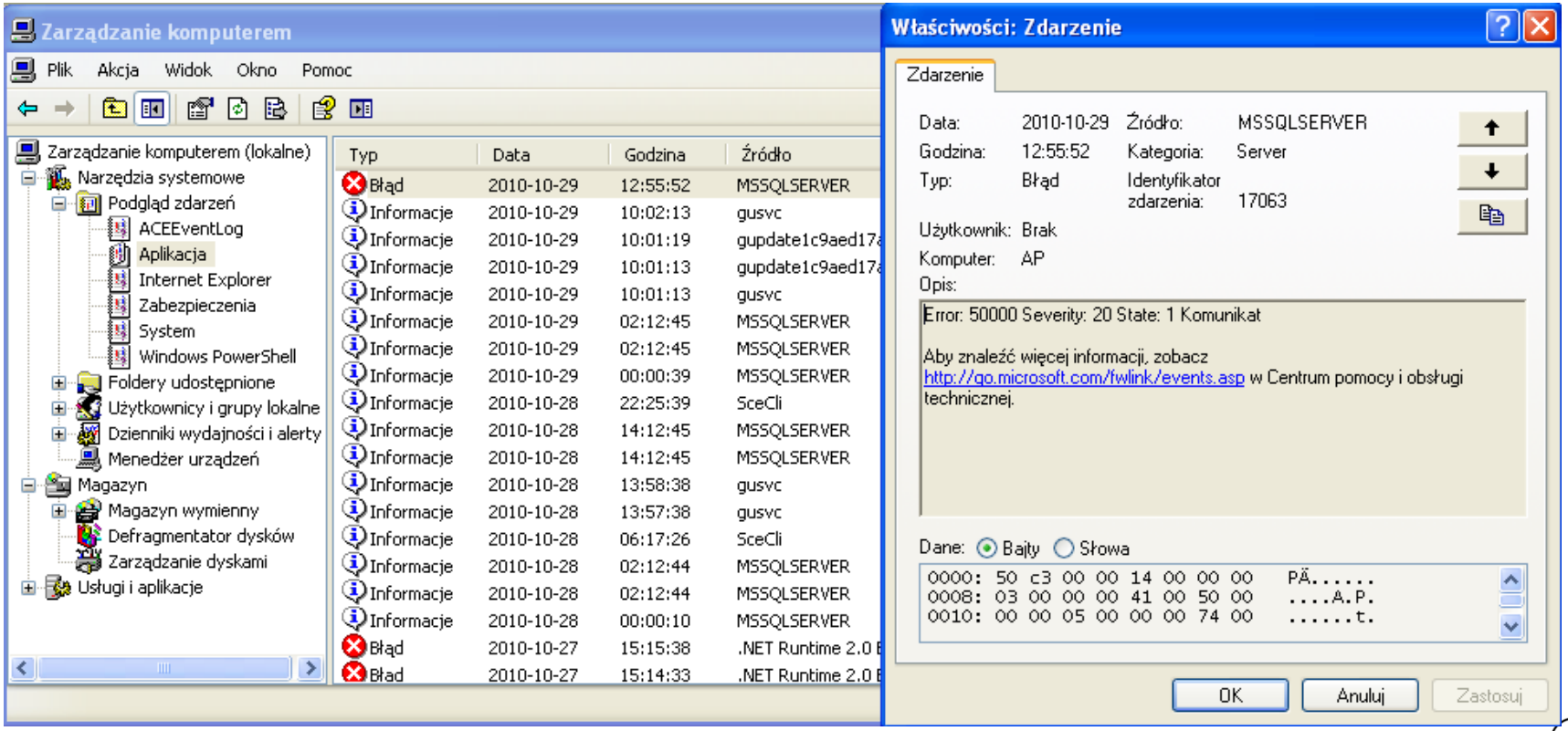
```
RAISERROR ('Komunikat',
19,
1)WITH LOG
```

-- grupa błędów.

Zastosowanie ustawienia błędu do wyświetlenia komunikatu

Msg 50000, Level 19, State 1, Line 1

Komunikat



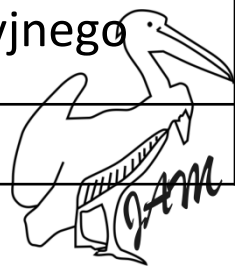
Błędy z opcją WITH LOG są widoczne w dzienniku systemu operacyjnego

0-18 zakres dostępny dla każdego użytkownika

19-25 tylko dla sysadmina lub użytkownika z uprawnieniami ALTER TRACE przy użyciu opcji WITH LOG (bledy od 20 do 25 traktowane są jako krytyczne)

Severity

Severity level	OPIS
0-9	Komunikaty informacyjne, silnik bazy danych nie powoduje ustawienia żadnego z komunikatów tej grupy
10	Komunikaty informacyjne, aby zachować zgodność wsteczną, silnik bazy danych konwertuje numer 10 do 0 przed zwróceniem do aplikacji, sam nie korzysta z tego numeru
11-16	Wskazuje błędy, których przyczyny mogą być skorygowane przez użytkownika
17-19	Wskazuje błędy programistyczne, których przyczyny mogą być skorygowane przez użytkownika, powiadamiając jednocześnie administratora
20-25	Wskazuje na błędy systemu oraz błędy krytyczne silnika bazy danych, bieżące zadania są przerywane, naprawa wymaga interwencji administratora, jest zapisywane do dziennika systemu operacyjnego
>25	Ustawiane są na wartość 25



Zastosowanie ustawienia błędu do wyświetlenia komunikatu

```
DECLARE @DBID INT;
SET @DBID = DB_ID();
DECLARE @DBNAME NVARCHAR(128);
SET @DBNAME = DB_NAME();
RAISERROR (N' Identyfikator bazy ID =%d,
Nazwa bazy = %s.',
10, -- grupa błędów.
1, -- podgrupa.
@DBID, -- Pierwszy argument.
@DBNAME); -- Drugi argument.
GO
```

Format	Opis
d lub i	Rzeczywista lub całkowita ze znakiem
o	Ósemkowa bez znaku
s	Łańcuch
u	Całkowita bez znaku
x lub X	szesnastkowa bez znaku

Tworzenie błędów użytkownika

```
sp_dropmessage 50001
GO
--Tworzenie błędu użytkownika
sp_addmessage 50001, 15, 'Komunikat';
GO
RAISERROR (50001,10,2)
GO
--Nadpisanie definicji błędu i uzupełnienie jej o parametry
sp_addmessage 50001, 15, 'Komunikat %d, %s', @replace='replace';
GO
DECLARE @i int, @s varchar(15)
SET @i=11
SET @s='napis'
RAISERROR (50001,10,2, @i, @s)
```

Komunikat
Komunikat 11, napis

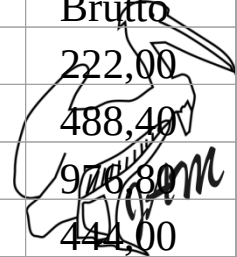
Transakcje – punkt wycofania transakcji SAVE

IdOsoby	Brutto
3	222,00
2	444,00
2	888,00
3	444,00

```
BEGIN TRANSACTION podwyzka
SELECT IdOsoby, Brutto FROM Zarobki WHERE IdOsoby IN (2, 3)
UPDATE Zarobki SET Brutto = Brutto*1.1 WHERE IdOsoby = 2

SAVE TRANSACTION podwyzka

UPDATE Zarobki SET brutto = brutto*0.9 WHERE idosoby = 3
SELECT IdOsoby, Brutto FROM Zarobki WHERE IdOsoby= 3

ROLLBACK TRANSACTION podwyzka
COMMIT TRANSACTION

SELECT IdOsoby,  Brutto FROM Zarobki WHERE IdOsoby IN (2, 3)
```

IdOsoby	Brutto
3	199,80
3	399,60

IdOsoby	Brutto
3	222,00
2	488,40
2	976,80
3	444,00

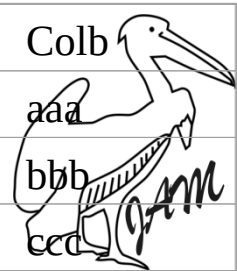


Transakcje hierarchiczne – (zagnieżdżone)

```
CREATE TABLE TestTran (Cola INT PRIMARY KEY, Colb CHAR(3))
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
GO
BEGIN TRANSACTION OuterTran
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (1, 'aaa')
GO
BEGIN TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (2, 'bbb')
GO
BEGIN TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (3, 'ccc')
GO
COMMIT TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
COMMIT TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
COMMIT TRANSACTION OuterTran
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
GO
SELECT * FROM TestTran
DROP TABLE TestTran
```

```
Transakcja 0
Transakcja 1
(1 row(s) affected)
Transakcja 2
(1 row(s) affected)
Transakcja 3
(1 row(s) affected)
Transakcja 2
Transakcja 1
Transakcja 0
(3 row(s) affected)
```

Cola	Colb
1	aaa
2	bbb
3	ccc



Transakcje hierarchiczne – (zagnieżdżone)

```
CREATE TABLE TestTran (Cola INT PRIMARY KEY, Colb CHAR(3))
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
GO
BEGIN TRANSACTION OuterTran
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (1, 'aaa')
GO
BEGIN TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (2, 'bbb')
GO
BEGIN TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (3, 'ccc')
GO
COMMIT TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
COMMIT TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
ROLBACK TRANSACTION OuterTran
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
GO
SELECT * FROM TestTran
DROP TABLE TestTran
```

```
Transakcja 0
Transakcja 1
(1 row(s) affected)
Transakcja 2
(1 row(s) affected)
Transakcja 3
(1 row(s) affected)
Transakcja 2
Transakcja 1
Transakcja 0
(3 row(s) affected)
```

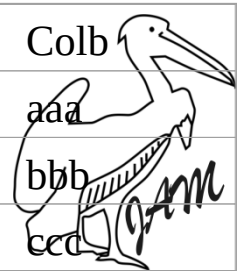
Cola	Colb

Transakcje hierarchiczne – (zagnieżdżone)

```
CREATE TABLE TestTran (Cola INT PRIMARY KEY, Colb CHAR(3))
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
GO
BEGIN TRANSACTION OuterTran
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (1, 'aaa')
GO
BEGIN TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (2, 'bbb')
GO
BEGIN TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (3, 'ccc')
GO
ROLBACK TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
COMMIT  TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
COMMIT  TRANSACTION OuterTran
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
GO
SELECT * FROM TestTran
DROP TABLE TestTran
```

```
Transakcja 0
Transakcja 1
(1 row(s) affected)
Transakcja 2
(1 row(s) affected)
Transakcja 3
(1 row(s) affected)
Msg 6401, Level 16, State 1, Line 1
Cannot roll back Inner2. No
transaction or savepoint of that
name was found.
Transakcja 3
Transakcja 2
Transakcja 1
(3 row(s) affected)
```

Cola	Colb
1	aaa
2	bbb
3	ccc

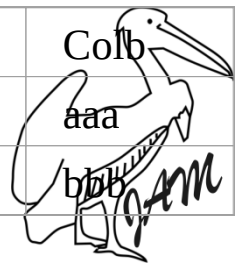


Transakcje hierarchiczne – (zagnieżdżone)

```
CREATE TABLE TestTran (Cola INT PRIMARY KEY, Colb CHAR(3))
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
GO
BEGIN TRANSACTION OuterTran
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (1, 'aaa')
GO
BEGIN TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (2, 'bbb')
GO
BEGIN TRANSACTION Inner2
SAVE TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (3, 'ccc')
GO
ROLBACK TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
COMMIT  TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
COMMIT  TRANSACTION OuterTran
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
GO
SELECT * FROM TestTran
DROP TABLE TestTran
```

```
Transakcja 1
Transakcja 2
(1 row(s) affected)
Transakcja 3
(1 row(s) affected)
Transakcja 4
(1 row(s) affected)
Transakcja 4
Transakcja 3
Transakcja 2
(2 row(s) affected)
```

Cola	Colb
1	aaa
2	bbb



Transakcje hierarchiczne – (zagnieżdżone)

```
CREATE TABLE TestTran (Cola INT PRIMARY KEY, Colb CHAR(3))
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
GO
BEGIN TRANSACTION OuterTran
SAVE TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (1, 'aaa')
GO
BEGIN TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (2, 'bbb')
GO
BEGIN TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (3, 'ccc')
GO
ROLBACK TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
COMMIT  TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
COMMIT  TRANSACTION OuterTran
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
GO
SELECT * FROM TestTran
DROP TABLE TestTran
```

```
Transakcja 2
Transakcja 3
(1 row(s) affected)
Transakcja 4
(1 row(s) affected)
Transakcja 5
(1 row(s) affected)
Transakcja 5
Transakcja 4
Transakcja 3
(0 row(s) affected)
```

Cola	Colb

Transakcje hierarchiczne – (zagnieżdżone)

```
CREATE TABLE TestTran (Cola INT PRIMARY KEY, Colb CHAR(3))
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
GO
BEGIN TRANSACTION OuterTran
SAVE TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (1, 'aaa')
GO
BEGIN TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (2, 'bbb')
GO
BEGIN TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (3, 'ccc')
GO
ROLBACK TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
COMMIT  TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
COMMIT  TRANSACTION OuterTran
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
GO
SELECT * FROM TestTran
DROP TABLE TestTran
WHILE @@TRANCOUNT >0
COMMIT TRAN
```

```
Transakcja 0
Transakcja 1
(1 row(s) affected)
Transakcja 2
(1 row(s) affected)
Transakcja 3
(1 row(s) affected)
Transakcja 2
Transakcja 1
Transakcja 0
(3 row(s) affected)
```

Aby zamknąć wszystkie transakcje, które nie są ani zatwierdzone ani wycofane

Cola	Colb

Transakcje hierarchiczne – (zagnieżdżone)

```
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
BEGIN TRANSACTION OuterTran
CREATE TABLE TestTran (Cola INT PRIMARY KEY, Colb CHAR(3))
GO
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (1, 'aaa')
GO
BEGIN TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (2, 'bbb')
GO
BEGIN TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
INSERT INTO TestTran VALUES (3, 'ccc')
GO
COMMIT TRANSACTION Inner2
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
COMMIT TRANSACTION Inner1
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
ROLLBACK TRANSACTION OuterTran
PRINT 'Transakcja '+ CAST(@@TRANCOUNT AS varchar(2))
GO
SELECT * FROM TestTran
DROP TABLE TestTran
WHILE @@TRANCOUNT >0
```

```
Transakcja 0
Transakcja 1
(1 row(s) affected)
Transakcja 2
(1 row(s) affected)
Transakcja 3
(1 row(s) affected)
Transakcja 2
Transakcja 1
Transakcja 0
Msg 208, Level 16, State 1, Line 1
Invalid object name 'TestTran'.
```

Blok TRY CATCH dzielenie przez zero

```
BEGIN TRY
   -- Dzielenie przez zero.
   SELECT 1/0;
END TRY
BEGIN CATCH
   SELECT
      ERROR_NUMBER() AS NumerBłędu,
      ERROR_SEVERITY() AS SeverityBłędu,
      ERROR_STATE() AS StateBłędu,
      ERROR_PROCEDURE() AS ProceduraBłędu,
      ERROR_LINE() AS LiniaBłędu,
      ERROR_MESSAGE() AS KomunikatBłędu;
END CATCH;
```

Numer Błędu	Severity Błędu	State Błędu	Procedura Błędu	Linia Błędu	KomunikatBłędu
8134	16	1	NULL	3	Divide by zero error encountered.

Blok TRY_CATCH bez wywołania procedury – nie przynosi skutku

```
BEGIN TRY
   -- Tabela nie istnieje błąd nie jest przechwytywany
SELECT * FROM NieIstniejaca;
END TRY
BEGIN CATCH
   SELECT
      ERROR_NUMBER() as ErrorNumber,
      ERROR_MESSAGE() as ErrorMessage;
END CATCH
```

Skutek

```
 Msg 208, Level 16, State 1, Line 4
Invalid object name ' NieIstniejaca '.
```

Procedura i jej wywołanie z TRY_CATCH

```
DROP PROCEDURE BladProc;
GO
CREATE PROCEDURE BladProc
AS
   SELECT * FROM NieIstniejaca;
GO
```

```
BEGIN TRY
   EXECUTE BladProc
END TRY
BEGIN CATCH
   SELECT
      ERROR_NUMBER() as
ErrorNumber,
      ERROR_MESSAGE() as
ErrorMessage;
END CATCH;
```

```
Skutek
ErrorNumber ErrorMessage
----------- -------------------------------------
208         Invalid object name 'NieIstniejaca'.

(1 row(s) affected)
```

Procedura i jej wywołanie z TRY_CATCH

```
DROP PROCEDURE BladProc;
GO
CREATE PROCEDURE BladProc
AS
   SELECT * FROM NieIstniejaca;
GO
```

Wywołanie proc bez try_catch kończy się błędem

```
EXEC BladProc;
```

```
Skutek
```

Msg 208, Level 16, State 1, Procedure BladProc, Line 3

Invalid object name 'NieIstniejaca'.

Próba usunięcia powiązanych rekordów klucz obcy , w obrębie transakcji z zastosowaniem Try_catch

```
BEGIN TRANSACTION
BEGIN TRY
    -- Osoba o IdOsoby =2 ma zarobki
    DELETE FROM Osoby WHERE IdOsoby=2
END TRY
BEGIN CATCH
    SELECT ERROR_NUMBER() AS ErrorNumber,
            ERROR_SEVERITY() AS ErrorSeverity,
            ERROR_STATE() as ErrorState,
            ERROR_PROCEDURE() as ErrorProcedure,
            ERROR_LINE() as ErrorLine,
            ERROR_MESSAGE() as ErrorMessage
    IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION
END CATCH
IF @@TRANCOUNT > 0
    COMMIT TRANSACTION
```

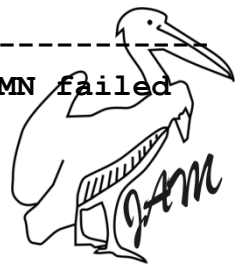
Skutek

```
ErrorNumber ErrorSeverity ErrorState  ErrorProcedure   ErrorLine   ErrorMessage
----------- ------------- ----------- ---------------- ----------- ------------------------------
547         16            0           NULL             5           The DELETE statement
conflicted with the REFERENCE constraint "FK_Zarobki_Osoby". The conflict occurred in database
"ap", table "dbo.Zarobki", column 'IdOsoby'.

(1 row(s) affected)
```

Próba usunięcia niejstniejącej kolumny kod z tabeli Dzialy

```
BEGIN TRANSACTION
BEGIN TRY
    -- Generowany jest błąd z powodu nieistnienia usuwanej kolumny
    ALTER TABLE dzialy DROP COLUMN kod;
   -- Jeśli polecenie powiedzie się zatwierdź transakcje.
    COMMIT TRANSACTION;
END TRY
BEGIN CATCH
   SELECT ERROR_NUMBER() AS ErrorNumber,
            ERROR_SEVERITY() AS ErrorSeverity,
            ERROR_STATE() as ErrorState,
            ERROR_PROCEDURE() as ErrorProcedure,
            ERROR_LINE() as ErrorLine,
            ERROR_MESSAGE() as ErrorMessage
    IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION
END CATCH
IF @@TRANCOUNT > 0
    COMMIT TRANSACTION
```

```
ErrorNumber ErrorSeverity ErrorState  ErrorProcedure  ErrorLine   ErrorMessage
----------- ------------- ---------- --------------- ---------- ------------------------------
4924        16            1          NULL            4          ALTER TABLE DROP COLUMN failed
because column 'kod' does not exist in table 'Dzialy'.

(1 row(s) affected)
```



Wymuszenie błędu w TRY CATCH z zastosowaniem RAISERROR

```
BEGIN TRY
        -- RAISERROR z grup 11-19 przeniesie wykonanie do bloku CATCH
        -- Niższe grupy nie powodują przeniesienia
        RAISERROR ('Błąd w bloku TRY .', 16, 1 )
END TRY
BEGIN CATCH
        DECLARE @ErrorMessage NVARCHAR(4000)
        DECLARE @ErrorSeverity INT
        DECLARE @ErrorState INT
        SELECT @ErrorMessage = 'Z CATCH ' + ERROR_MESSAGE(),
      @ErrorSeverity = ERROR_SEVERITY(),
      @ErrorState = ERROR_STATE()
   -- zastosowano RAISERROR w bloku CATCH aby zwrócić
   -- informacje o błędzie, który spowodował przeniesienie
   -- wykonywania skryptu do bloku  CATCH .
   RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState)
END CATCH
```

Dla ErrorSeverity **>10**

Msg 50000, Level 16, State 1, Line 16

Z CATCH Błąd w bloku TRY .

Dla ErrorSeverity **<=10**

Błąd w bloku TRY .

TRY CATCH

```
DROP TABLE Test
GO
CREATE TABLE Test
(IdTest int
IDENTITY(1,1)
PRIMARY KEY,
wart int)
GO
```

```
BEGIN TRANSACTION
BEGIN TRY
        INSERT INTO test VALUES(11)
        SELECT * FROM TEST
        INSERT INTO test VALUES('22')
        SELECT * FROM TEST
        INSERT INTO test VALUES('aa')
        SELECT * FROM TEST
        INSERT INTO test VALUES(44)
        SELECT * FROM TEST
        COMMIT TRANSACTION;
END TRY
BEGIN CATCH
   SELECT ERROR_NUMBER() AS ErrorNumber,
                ERROR_SEVERITY() AS ErrorSeverity,
                ERROR_STATE() as ErrorState,
                ERROR_PROCEDURE() as ErrorProcedure,
                ERROR_LINE() as ErrorLine,
                ERROR_MESSAGE() as ErrorMessage
        IF @@TRANCOUNT > 0
                ROLLBACK TRANSACTION
END CATCH
IF @@TRANCOUNT > 0
        COMMIT TRANSACTION
        SELECT * FROM TEST
```

```
IdTest      wart
----------- -----------
1           11

IdTest      wart
----------- -----------
1           11
2           22

ErrorNumber ErrorSeverity ErrorState
ErrorProcedure
ErrorLine   ErrorMessage

245         16            1          NULL
7           Conversion failed when
converting the varchar value 'aa' to data
type int.

IdTest      wart
----------- -----------
```

TRY CATCH w procedurze

```
CREATE PROCEDURE wsp
@dzial int=1,
@ile real OUTPUT
AS
SELECT @ile=MAX(Wzrost)/(MAX(Wzrost)-MIN(Wzrost))
FROM Osoby WHERE IdDzialu=@dzial
GO
DECLARE @ile real
EXEC wsp 9, @ile OUTPUT
PRINT @ile
```

Msg 8134, Level 16, State 1,
Procedure wsp, Line 5
Divide by zero error encountered.

```
CREATE PROCEDURE wsp
@dzial int=1,
@ile real OUTPUT
AS
BEGIN TRY
SELECT @ile=MAX(Wzrost)/(MAX(Wzrost)-
MIN(Wzrost))
FROM Osoby WHERE IdDzialu=@dzial
END TRY
BEGIN CATCH
SET @ile=NULL
END CATCH
GO
DECLARE @ile real
EXEC wsp 9, @ile OUTPUT
PRINT @ile
```

Następnie Indeksy