

RAISERROR

TRANSACTIONS AND ERROR HANDLING IN SQL SERVER



Miriam Antona
Software Engineer

Raise errors statements

- `RAISERROR`
- `THROW`
- Microsoft suggests `THROW`

RAISERROR syntax

```
RAISERROR ( { msg_str | msg_id | @local_variable_message },  
           severity,  
           state,  
           [ argument [ ,...n ] ] )  
[ WITH option [ ,...n ] ]
```

RAISERROR with message string

```
IF NOT EXISTS (SELECT * FROM staff WHERE staff_id = 15)
    RAISERROR('No staff member with such id.', 16, 1);
```

```
Msg. 50000, Level 16, State 1, Line 3
No staff member with such id.
```

RAISERROR with message string

```
IF NOT EXISTS (SELECT * FROM staff WHERE staff_id = 15)
    RAISERROR('No %s with id %d.', 16, 1, 'staff member', 15);
```

```
Msg. 50000, Level 16, State 1, Line 3
No staff member with id 15.
```

```
RAISERROR('%d%% discount', 16, 1, 50);
```

```
Msg. 50000, Level 16, State 1, Line 1
50% discount
```

- Other characters: %i, %o, %x, %X, %u...

RAISERROR with error number

```
RAISERROR(60000, 16, 1);
```

```
Msg. 60000, Level 16, State 1, Line 1  
This is a test message.
```

```
SELECT * FROM sys.messages
```

message_id	language_id	severity	is_event_logged	text
...
60000	1033	16	0	This is a test message
...

RAISERROR - Example with TRY...CATCH

```
BEGIN TRY
    IF NOT EXISTS (SELECT * FROM staff WHERE staff_id = 15)
        RAISERROR('No staff member with such id.', 9, 1);
END TRY
BEGIN CATCH
    SELECT 'You are in the CATCH block' AS message
END CATCH
```

```
No staff member with such id.
Msg. 50000, Level 9, State 1
```

RAISERROR - Example with TRY...CATCH

```
BEGIN TRY
    IF NOT EXISTS (SELECT * FROM staff WHERE staff_id = 15)
        RAISERROR('No staff member with such id.', 16, 1);
END TRY
BEGIN CATCH
    SELECT 'You are in the CATCH block' AS message
END CATCH
```

```
| message |
|-----|
| You are in the CATCH block |
```


Let's practice!

TRANSACTIONS AND ERROR HANDLING IN SQL SERVER

THROW

TRANSACTIONS AND ERROR HANDLING IN SQL SERVER



Miriam Antona
Software Engineer

THROW syntax

- Recommended by Microsoft over the `RAISERROR` statement.

```
THROW [ error_number, message, state ][ ; ]
```

THROW - without parameters

```
BEGIN TRY
    SELECT price/0 from orders;
END TRY
BEGIN CATCH
    THROW;
    SELECT 'This line is executed!' as message;
END CATCH
```

```
(0 rows affected)
Msg. 8134, Level 16, State 1, Line 2
Divide by zero error encountered.
```

THROW - ambiguity

```
BEGIN TRY
    SELECT price/0 from orders;
END TRY
BEGIN CATCH
    SELECT 'This line is executed!'
    THROW;
END CATCH
```

```
| THROW |
|-----|
| This line is executed! |
```

THROW - ambiguity

```
BEGIN TRY
    SELECT price/0 from orders;
END TRY
BEGIN CATCH
    SELECT 'This line is executed!';
    THROW;
END CATCH
```

```
| (No column name) |
|-----|
| This line is executed! |
```

```
(0 rows affected)
(1 rows affected)
Msg. 8134, Level 16, State 1, Line 2
Divide by zero error encountered.
```

THROW - with parameters

```
THROW error_number, message, state [ ; ]
```

```
THROW 52000, 'This is an example', 1;
```

```
Msg. 52000, Level 16, State 1, Line 1  
This is an example
```

THROW - with parameters

```
BEGIN TRY
    IF NOT EXISTS (SELECT * FROM staff WHERE staff_id = 15)
        THROW 51000, 'This is an example', 1;
END TRY
BEGIN CATCH
    SELECT ERROR_MESSAGE() AS message;
END CATCH
```

```
| THROW |
|-----|
| This is an example |
```


Let's practice!

TRANSACTIONS AND ERROR HANDLING IN SQL SERVER

Customizing error messages in the **THROW** statement

TRANSACTIONS AND ERROR HANDLING IN SQL SERVER

SQL

Miriam Antona
Software Engineer

Parameter placeholders in RAISERROR and THROW

```
RAISERROR('No %s with id %d.', 16, 1, 'staff member', 15);
```

```
THROW 52000, 'No staff member with id 15', 1;
```

Ways of customizing error messages

- Variable by concatenating strings
- `FORMATMESSAGE` function

Using a variable and the CONCAT function

```
DECLARE @staff_id AS INT = 500;  
DECLARE @my_message NVARCHAR(500) =  
    CONCAT('There is no staff member for id ', @staff_id, '. Try with another one.');
```



```
IF NOT EXISTS (SELECT * FROM staff WHERE staff_id = @staff_id)  
    THROW 50000, @my_message, 1;
```

```
Msg. 50000, Level 16, State 1, Line 5  
There is no staff member for id 500. Try with another one.
```

The FORMATMESSAGE function

```
FORMATMESSAGE ( { ' msg_string ' | msg_number } ,  
                [ param_value [ ,...n ] ] )
```

FORMATMESSAGE with message string

```
DECLARE @staff_id AS INT = 500;
DECLARE @my_message NVARCHAR(500) =
    FORMATMESSAGE('There is no staff member for id %d. %s ', @staff_id, 'Try with another one.');
```



```
IF NOT EXISTS (SELECT * FROM staff WHERE staff_id = @staff_id)
    THROW 50000, @my_message, 1;
```

```
Msg. 50000, Level 16, State 1, Line 6
There is no staff member for id 500. Try with another one.
```

FORMATMESSAGE with message number

```
SELECT * FROM sys.messages
```

message_id	language_id	severity	is_event_logged	text
60000	1033	16	0	This is a test message with one numeric parameter (%d)...
21	1033	20	0	Warning: Fatal error %d occurred at %S_DATE. Note the error...
101	1033	15	0	Query not allowed in Waitfor.
102	1033	15	0	Incorrect syntax near '%.*ls'.
103	1033	15	0	The %S_MSG that starts with '%.*ls' is too long...
...

FORMATMESSAGE with message number

```
sp_addmessage  
    msg_id , severity , msgtext,  
    [ language ],  
    [ with_log { 'TRUE' | 'FALSE' } ],  
    [ replace ]
```

- msg_id > 50000

FORMATMESSAGE with message number

```
exec sp_addmessage
    @msgnum = 55000, @severity = 16, @msgtext = 'There is no staff member for id %d. %s', @lang = N'us_english';
```

message_id	language_id	severity	is_event_logged	text
55000	1033	16	0	There is no staff member for id %d. %s

```
DECLARE @staff_id AS INT = 500;
DECLARE @my_message NVARCHAR(500) = FORMATMESSAGE(55000, @staff_id, 'Try with another one.');
```

```
IF NOT EXISTS (SELECT * FROM staff WHERE staff_id = @staff_id)
    THROW 50000, @my_message, 1;
```

```
Msg. 50000, Level 16, State 1, Line 6
There is no staff member for id 500. Try with another one.
```

Let's practice!

TRANSACTIONS AND ERROR HANDLING IN SQL SERVER