# Examining Error Handling Methods

**Jared Westover**
SQL ARCHITECT

@WestoverJared

## Module Overview

**Examining an error message**

- Number, message, and severity
- Which errors require attention
- RAISERROR command

**Implementing TRY..CATCH blocks**

- Passing control
- When control is not passed
- Using SET XACT_ABORT

**Using THROW**

- General overview
- Main benefits

## Examining an Error Message

Number

Severity

Line

Message

State

Procedure

```
SELECT 1/0;
GO
Msg 8134, Level 16, State 1, Line 1
Divide by zero error encountered.
```

## Error Message Details

**Did you know there are over 40,000 messages in SQL Server**

## Errors Requiring Attention

**Error severity levels**

- 0-9 informational
- 11-16 user can fix
- 17-19 resource issues
- 20-25 fatal errors

**Focus on 11-16 since we can fix them**

- Foreign key violations
- Data type conversion

## RAISERROR

**Only command used to raise exceptions before SQL 2012**

**Must pass in required parameters**

- Message, severity, and state
- Default id of 50,000

**Raise informational messages**

- Unexpected counts

**With log option**

## Demo

**Explore the SQL error log**

**Examine error messages**

- Line number

**Where are messages stored**

- Adding a new user defined message

Demo

Using RAISERROR to raise an exception

- With log

## TRY..CATCH

**Exception handling**

- Simplified the process
- Any chance of an error

**Passing the control**

- Severity level greater than 10

**Nested blocks**

- Similar to nested transactions

**Error functions**

- Only used inside of CATCH

```
BEGIN TRY

-- Do something amazing

END TRY

BEGIN CATCH

-- An error message occurred

END CATCH
```

## Syntax for TRY..CATCH

**Don’t forget to include the CATCH**

```
SET XACT_ABORT ON;
BEGIN TRANSACTION;
    CREATE TABLE #TestTable (Id int);
    SELECT 1/0;
COMMIT TRANSACTION;
GO
```

## Implementing XACT_ABORT

**Terminates all statements from the transaction**

```sql
IF (XACT_STATE()) = -1 -- This one is doomed

IF (XACT_STATE()) = 1 -- This one is committable
```

## Using XACT_STATE

**A function used to determine the state of the current transaction**

## Demo

Review TRY..CATCH examples

Using RAISERROR

Error functions

## Demo

Review SET XACT_ABORT ON behavior

Transaction state with XACT_STATE()

## THROW

**Released in SQL Server 2012**

- Microsoft recommends using

**Raise the actual error**

**No need for parameters**

**Only severity level 16**

**Remember the semicolon**

Demo

## Using THROW to raise an exception

- Accurate line number
- The actual message id

## What We Covered

**Error message components**

- Message id, message, severity, & state
- TRY..CATCH blocks

**Two methods for raising an exception**

- RAISERROR
- THROW

**Additional error handling details**

- XACT_ABORT
- XACT_STATE

Next Module: Handling Errors in the Real World