**CURSO T-SQL – ALURA:**

**SQL Server: T-SQL com Microsoft SQL Server 2017**

**<https://cursos.alura.com.br/course/tsql-com-sql-server-2017/task/41176>**

**05. TRATAMENTO DE ERROS**

Nesta aula, aprendemos:

* Como tratar erros no T-SQL;
* Como tratar erros pelas variáveis **@@ERROR** e **@@ROWCOUNT**;
* Como usar a estrutura de **TRY-CATCH** para tratar os erros;
* Algumas funções que auxiliam a criação de mensagens para o usuário, através do **TRY-CATCH**;
* Como usar o ***debug*** para verificar um problema durante a execução do problema desta aula.

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

1) Crie uma nova consulta no Management Studio. Execute a consulta abaixo, que vai achar o faturamento e calcular uma comissão, que é o valor do faturamento dividido por 10:

SELECT SUM(A.QUANTIDADE \* A.[PREÇO]) AS FATURAMENTO,

SUM(A.QUANTIDADE \* A.[PREÇO])/10 AS COMISSAO

FROM [ITENS NOTAS FISCAIS] A INNER JOIN [NOTAS FISCAIS] B

ON A.NUMERO = B.NUMERO WHERE B.CPF = '1471156710' AND YEAR(B.[DATA]) = 2016

2) Declare uma variável que fará o papel do denominador da divisão, usando na consulta acima, mas usando com o valor 0:

DECLARE @DENOMINADOR INT

SET @DENOMINADOR = 0

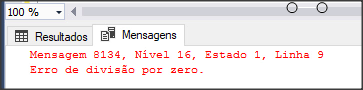
SELECT SUM(A.QUANTIDADE \* A.[PREÇO]) AS FATURAMENTO,

SUM(A.QUANTIDADE \* A.[PREÇO])/@DENOMINADOR AS COMISSAO

FROM [ITENS NOTAS FISCAIS] A INNER JOIN [NOTAS FISCAIS] B

ON A.NUMERO = B.NUMERO WHERE B.CPF = '1471156710' AND YEAR(B.[DATA]) = 2016

Executando esta consulta, acontecerá o erro abaixo:



3) Crie uma *stored procedure* que irá tratar o erro. Digite e execute:

CREATE PROCEDURE TrataErroZero

@CPF VARCHAR(12), @ANO INT, @DENOMINADOR INT,

@NUMERRO INT OUTPUT, @NUMLINHA INT OUTPUT

AS

BEGIN

SELECT SUM(A.QUANTIDADE \* A.[PREÇO]) AS FATURAMENTO,

SUM(A.QUANTIDADE \* A.[PREÇO])/@DENOMINADOR AS COMISSAO

FROM [ITENS NOTAS FISCAIS] A INNER JOIN [NOTAS FISCAIS] B

ON A.NUMERO = B.NUMERO

WHERE B.CPF = @CPF AND YEAR(B.[DATA]) = @ANO

SELECT @NUMERRO = @@ERROR, @NUMLINHA = @@ROWCOUNT

END

4) Chame a SP criada acima:

DECLARE @DENOMINADOR INT

DECLARE @CPF VARCHAR(12)

DECLARE @ANO INT

DECLARE @NUMERRO INT

DECLARE @NUMLINHA INT

SET @CPF = '1471156710'

SET @ANO = 2016

SET @DENOMINADOR = 0

EXEC TrataErroZero @CPF = @CPF, @ANO = @ANO,

@DENOMINADOR = @DENOMINADOR,

@NUMERRO = @NUMERRO OUTPUT,

@NUMLINHA = @NUMLINHA OUTPUT

5) Antes de executar os comandos, trate o erro que será retornado pela variável **@NUMLINHA** e **@NUMERRO**:

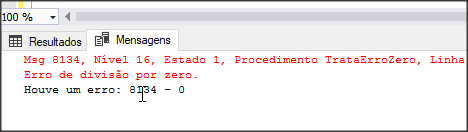
IF @NUMERRO <> 0

PRINT 'Houve um erro: ' +

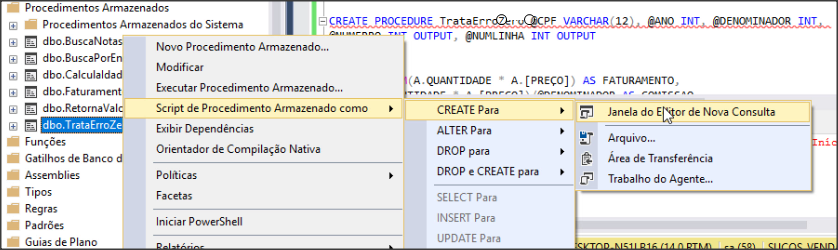
CONVERT(VARCHAR(30), @NUMERRO) + ' - ' +

CONVERT(VARCHAR(30), @NUMLINHA)

6) Executando a rotina completa, você terá:



7) Edite o exemplo anterior, mas agora usando a estrutura de tratamento de erros **TRY-CATCH**. Para isso, clique com o botão da direita do mouse sobre a rotina criada acima (**TrataErroZero)), que está salva na pasta**Programação/Procedimentos Armazenados**, e escolha a opção**Script de Procedimento Armazenado --> CREATE Para --> Janela do Editor de Nova Consulta\*\*:



8) Sobre a rotina que foi escrita pelo SQL Server, retire as linhas desnecessárias, de tal modo que você terá:

CREATE PROCEDURE TrataErroZero

@CPF VARCHAR(12), @ANO INT, @DENOMINADOR INT,

@NUMERRO INT OUTPUT, @NUMLINHA INT OUTPUT

AS

BEGIN

SELECT SUM(A.QUANTIDADE \* A.[PREÇO]) AS FATURAMENTO,

SUM(A.QUANTIDADE \* A.[PREÇO])/@DENOMINADOR AS COMISSAO

FROM [ITENS NOTAS FISCAIS] A INNER JOIN [NOTAS FISCAIS] B

ON A.NUMERO = B.NUMERO WHERE B.CPF = @CPF AND YEAR(B.[DATA]) = @ANO

SELECT @NUMERRO = @@ERROR, @NUMLINHA = @@ROWCOUNT

END

9) Modifique esta rotina para o uso do **TRY-CATCH**, você terá:

CREATE PROCEDURE [dbo].[TrataErroZeroTry]

@CPF VARCHAR(12), @ANO INT,

@DENOMINADOR INT, @MENSAGEM VARCHAR(50) OUTPUT

AS

BEGIN

BEGIN TRY

SELECT SUM(A.QUANTIDADE \* A.[PREÇO]) AS FATURAMENTO,

SUM(A.QUANTIDADE \* A.[PREÇO])/@DENOMINADOR AS COMISSAO

FROM [ITENS NOTAS FISCAIS] A

INNER JOIN [NOTAS FISCAIS] B

ON A.NUMERO = B.NUMERO WHERE B.CPF = @CPF

AND YEAR(B.[DATA]) = @ANO

END TRY

BEGIN CATCH

SET @MENSAGEM = 'Houve um erro número: ' +

CONVERT(VARCHAR(10), @@ERROR)

END CATCH

END

10) Execute os comandos acima para a criação da SP e depois digite:

DECLARE @DENOMINADOR INT

DECLARE @CPF VARCHAR(12)

DECLARE @ANO INT

DECLARE @MENSAGEM VARCHAR(30)

SET @CPF = '1471156710'

SET @ANO = 2016

SET @DENOMINADOR = 0

SET @MENSAGEM = ''

EXEC TrataErroZeroTry @CPF = @CPF,

@ANO = @ANO, @DENOMINADOR = @DENOMINADOR,

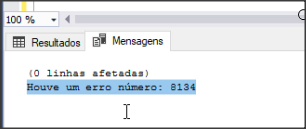
@MENSAGEM = @MENSAGEM OUTPUT

IF @MENSAGEM <> ''

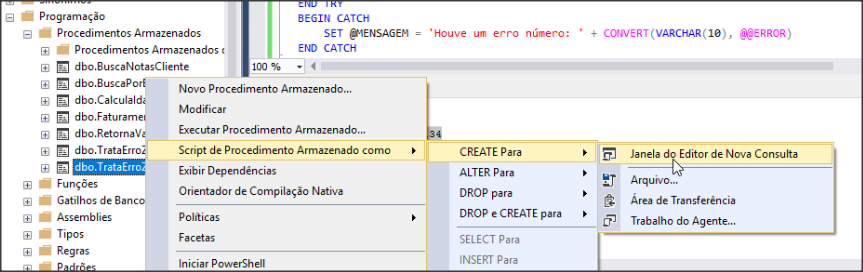
PRINT @MENSAGEM

Os comandos acima irão testar a rotina com **TRY-CATCH**. Também será passado zero como denominador.

11) Ao executar o teste, você terá:



12) Use as funções de tratamento de erro. Para isso, clique com o botão da direita do mouse sobre a rotina criada acima (**TrataErroZeroTry)), que está salva na pasta**Programação/Procedimentos Armazenados**, e escolha a opção**Script de Procedimento Armazenado --> CREATE Para --> Janela do Editor de Nova Consulta\*\*:



13) Limpe os códigos desnecessários e mantenha apenas o da criação da *procedure*:

CREATE PROCEDURE [dbo].[TrataErroZeroTry]

@CPF VARCHAR(12), @ANO INT, @DENOMINADOR INT,

@MENSAGEM VARCHAR(50) OUTPUT

AS

BEGIN

BEGIN TRY

SELECT SUM(A.QUANTIDADE \* A.[PREÇO]) AS FATURAMENTO,

SUM(A.QUANTIDADE \* A.[PREÇO])/@DENOMINADOR AS COMISSAO

FROM [ITENS NOTAS FISCAIS] A

INNER JOIN [NOTAS FISCAIS] B

ON A.NUMERO = B.NUMERO WHERE B.CPF = @CPF

AND YEAR(B.[DATA]) = @ANO

END TRY

BEGIN CATCH

SET @MENSAGEM = 'Houve um erro número: ' +

CONVERT(VARCHAR(10), @@ERROR)

END CATCH

END

14) Modifique o nome da *procedure* para **TrataErroZeroTry2**. Insira, dentro do **CATCH**, a exibição de algumas funções de tratamento de erro:

BEGIN CATCH

SET @MENSAGEM = 'Houve um erro número: ' +

CONVERT(VARCHAR(10), ERROR\_NUMBER()) + ' - '

SET @MENSAGEM = @MENSAGEM + 'Mensagem: ' +

ERROR\_MESSAGE() + ' - '

SET @MENSAGEM = @MENSAGEM + 'Grau de severidade do erro: ' +

CONVERT(VARCHAR(10), ERROR\_SEVERITY()) + ' - '

SET @MENSAGEM = @MENSAGEM + 'Estado do erro: ' +

CONVERT(VARCHAR(10), ERROR\_STATE()) + ' - '

SET @MENSAGEM = @MENSAGEM + 'Número da linha: ' +

CONVERT(VARCHAR(10), ERROR\_LINE()) + ' - '

SET @MENSAGEM = @MENSAGEM + 'Procedure: ' +

ERROR\_PROCEDURE()

END CATCH

15) A versão final do código da criação desta nova SP será:

CREATE PROCEDURE [dbo].[TrataErroZeroTry2]

@CPF VARCHAR(12), @ANO INT, @DENOMINADOR INT,

@MENSAGEM VARCHAR(50) OUTPUT

AS

BEGIN

BEGIN TRY

SELECT SUM(A.QUANTIDADE \* A.[PREÇO]) AS FATURAMENTO,

SUM(A.QUANTIDADE \* A.[PREÇO])/@DENOMINADOR AS COMISSAO

FROM [ITENS NOTAS FISCAIS] A

INNER JOIN [NOTAS FISCAIS] B

ON A.NUMERO = B.NUMERO WHERE B.CPF = @CPF

AND YEAR(B.[DATA]) = @ANO

END TRY

BEGIN CATCH

SET @MENSAGEM = 'Houve um erro número: ' +

CONVERT(VARCHAR(10), ERROR\_NUMBER()) + ' - '

SET @MENSAGEM = @MENSAGEM + 'Mensagem: ' +

ERROR\_MESSAGE() + ' - '

SET @MENSAGEM = @MENSAGEM + 'Grau de severidade do erro: ' +

CONVERT(VARCHAR(10), ERROR\_SEVERITY()) + ' - '

SET @MENSAGEM = @MENSAGEM + 'Estado do erro: ' +

CONVERT(VARCHAR(10), ERROR\_STATE()) + ' - '

SET @MENSAGEM = @MENSAGEM + 'Número da linha: ' +

CONVERT(VARCHAR(10), ERROR\_LINE()) + ' - '

SET @MENSAGEM = @MENSAGEM + 'Procedure: ' +

ERROR\_PROCEDURE()

END CATCH

END

16) Crie a SP executando os comandos acima e depois execute os comandos abaixo:

DECLARE @DENOMINADOR INT

DECLARE @CPF VARCHAR(12)

DECLARE @ANO INT

DECLARE @MENSAGEM VARCHAR(MAX)

SET @CPF = '1471156710'

SET @DENOMINADOR = 0

SET @ANO = 2016

SET @MENSAGEM = ''

EXEC TrataErroZeroTry2 @CPF = @CPF, @ANO = @ANO,

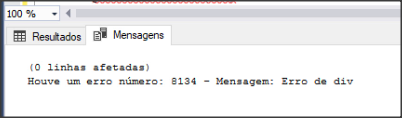
@DENOMINADOR = @DENOMINADOR,

@MENSAGEM = @MENSAGEM OUTPUT

IF @MENSAGEM <> ''

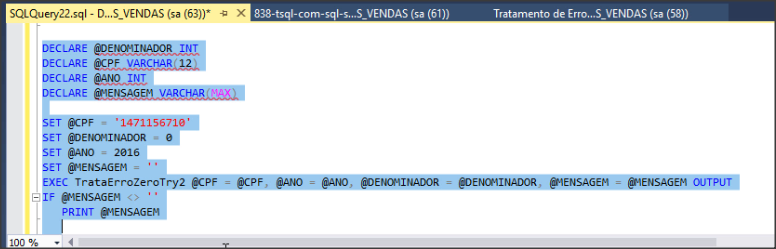
PRINT @MENSAGEM

17) Note que a mensagem de erro foi exibida, como mostrada abaixo:

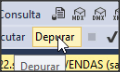


Houve um corte na mensagem. A princípio não se consegue identificar o porquê. Por isso use o **DEBUG** para achar o problema.

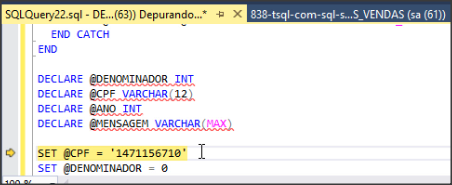
18) Selecione as linhas de execução da chamada da SP:

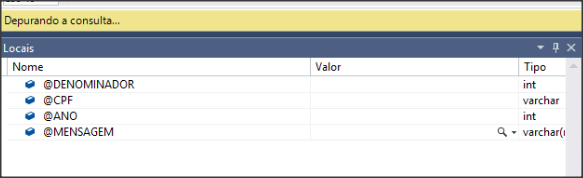


19) Clique no botão **Depurar**:



20) O depurador será apresentado:





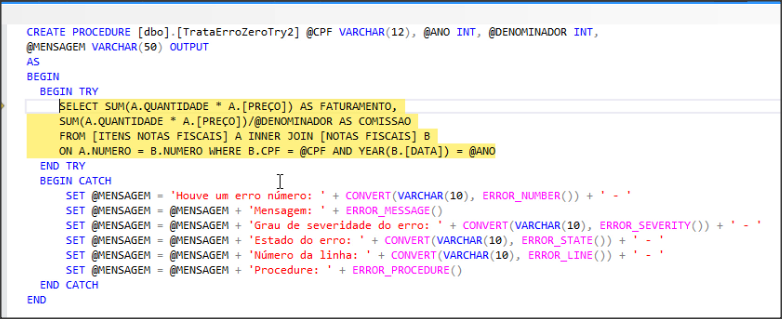
21) Clique no botão **Pular o método** até a chamada da SP:



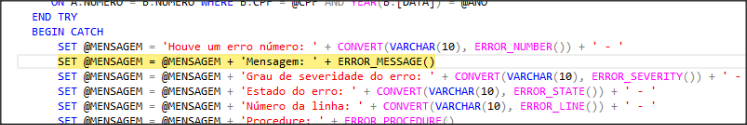
22) Você pode então entrar dentro da SP para depurar seu código. Para isso, clique no botão **Intervir**:



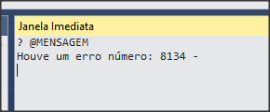
23) Assim você entrará dentro do código da SP:



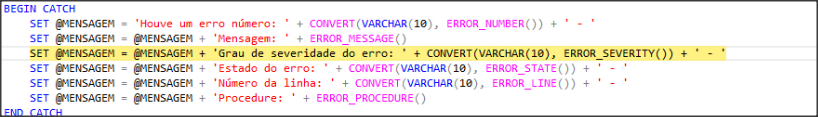
24) Após a escrita da mensagem na linha destacada abaixo:

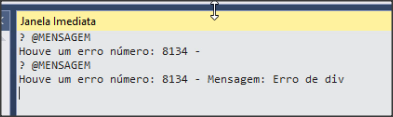


Você pode ir na janela ao lado (**Janela Imediata**) e digitar **?** mais o nome da variável (**@MENSAGEM**):



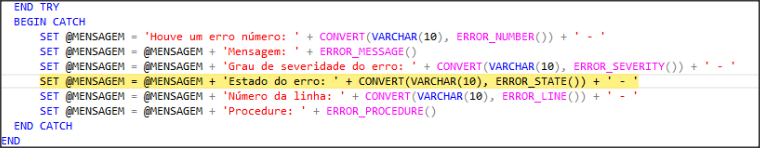
25) Indo para a próxima linha, você tem:





Note que a mensagem é exibida cortada.

26) Se você for para a próxima linha, a mensagem continua cortada:





27) Isso mostra que a variável **@MENSAGEM** tem um limite de tamanho. Isso porque na declaração da variável, foi limitado o tamanho para **VARCHAR(50)**:



28) Altere a SP, digitando o comando **ALTER PROCEDURE**, colocando a variável como **VARCHAR(MAX)**, que é a representação de um tamanho muito grande para o texto:

ALTER PROCEDURE [dbo].[TrataErroZeroTry2]

@CPF VARCHAR(12), @ANO INT, @DENOMINADOR INT,

@MENSAGEM VARCHAR(MAX) OUTPUT

AS

BEGIN

BEGIN TRY

SELECT SUM(A.QUANTIDADE \* A.[PREÇO]) AS FATURAMENTO,

SUM(A.QUANTIDADE \* A.[PREÇO])/@DENOMINADOR AS COMISSAO

FROM [ITENS NOTAS FISCAIS] A

INNER JOIN [NOTAS FISCAIS] B

ON A.NUMERO = B.NUMERO

WHERE B.CPF = @CPF AND YEAR(B.[DATA]) = @ANO

END TRY

BEGIN CATCH

SET @MENSAGEM = 'Houve um erro número: ' +

CONVERT(VARCHAR(10), ERROR\_NUMBER()) + ' - '

SET @MENSAGEM = @MENSAGEM + 'Mensagem: ' +

ERROR\_MESSAGE() + ' - '

SET @MENSAGEM = @MENSAGEM + 'Grau de severidade do erro: ' +

CONVERT(VARCHAR(10), ERROR\_SEVERITY()) + ' - '

SET @MENSAGEM = @MENSAGEM + 'Estado do erro: ' +

CONVERT(VARCHAR(10), ERROR\_STATE()) + ' - '

SET @MENSAGEM = @MENSAGEM + 'Número da linha: ' +

CONVERT(VARCHAR(10), ERROR\_LINE()) + ' - '

SET @MENSAGEM = @MENSAGEM + 'Procedure: ' +

ERROR\_PROCEDURE()

END CATCH

END

29) Execute os comandos para alteração da SP e depois execute-a novamente. Você terá agora a resposta correta:

