



Aula 18 – Introdução T-SQL

T-SQL

- No T-SQL podemos criar variáveis e elas podem ter valores.
- No T-SQL temos controle de fluxo e de decisão (if e while);
- Estrutura de tabelas temporárias;
- Tratamento de erro;
- Estrutura de cursores;
- Estruturas estas que podem ser salvas no SQL Server através de funções ou storage procedures.

T-SQL (Transact SQL)

- É Uma linguagem interpretada usada pelo SQL Server;
- Todos os comandos vistos até agora são T-SQL

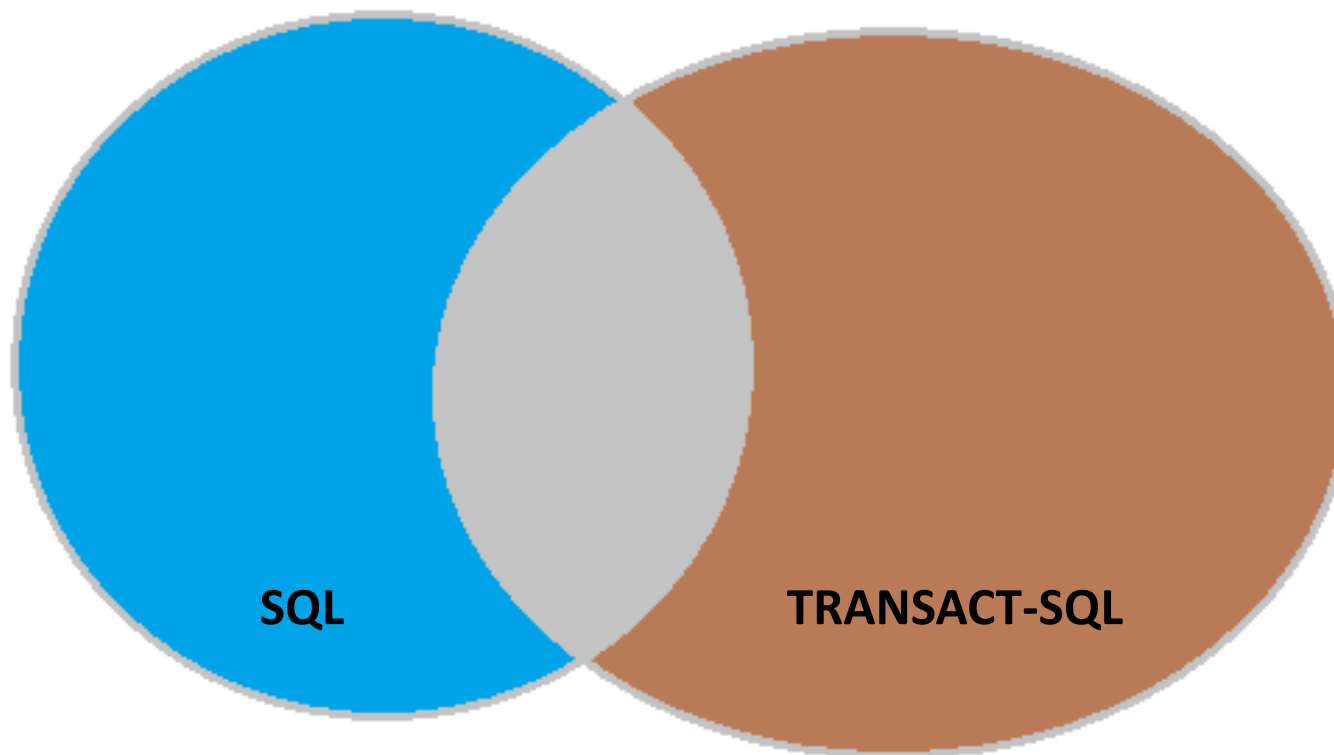
```
SELECT EMBALAGEM, TAMANHO FROM [TABELA DE PRODUTOS]
```

```
SELECT DISTINCT EMBALAGEM, TAMANHO FROM [TABELA DE PRODUTOS]
```

```
SELECT DISTINCT EMBALAGEM, TAMANHO FROM [TABELA DE  
PRODUTOS]
```

```
WHERE SABOR = 'Laranja'
```

T-SQL (Transact SQL)



T-SQL (Transact SQL)

- Com a programação T-SQL podemos expandir a manutenção do banco de dados;
- Podemos usar variáveis para criar uma estrutura dinâmica de atualização de dados.

```
DECLARE @TESTE VARCHAR (20)
```

```
SET @TESTE = 'SQL SERVER'
```

```
SELECT @TESTE AS 'UM TESTE'
```

T-SQL (Transact SQL)

```
DECLARE @RESPOSTA VARCHAR (20)
```

```
DECLARE @VALOR INT
```

```
SET @VALOR = 3
```

```
IF @VALOR = 3
```

```
    PRINT 'Valor menor ou igual a 10'
```

```
ELSE
```

```
    PRINT 'Valor maior que 10'
```

T-SQL (Transact SQL)

BEGIN

```
SELECT * FROM [TABELA DE NOTAS FISCAIS];  
SELECT * FROM [TABELA DE VENDEDORES];
```

END

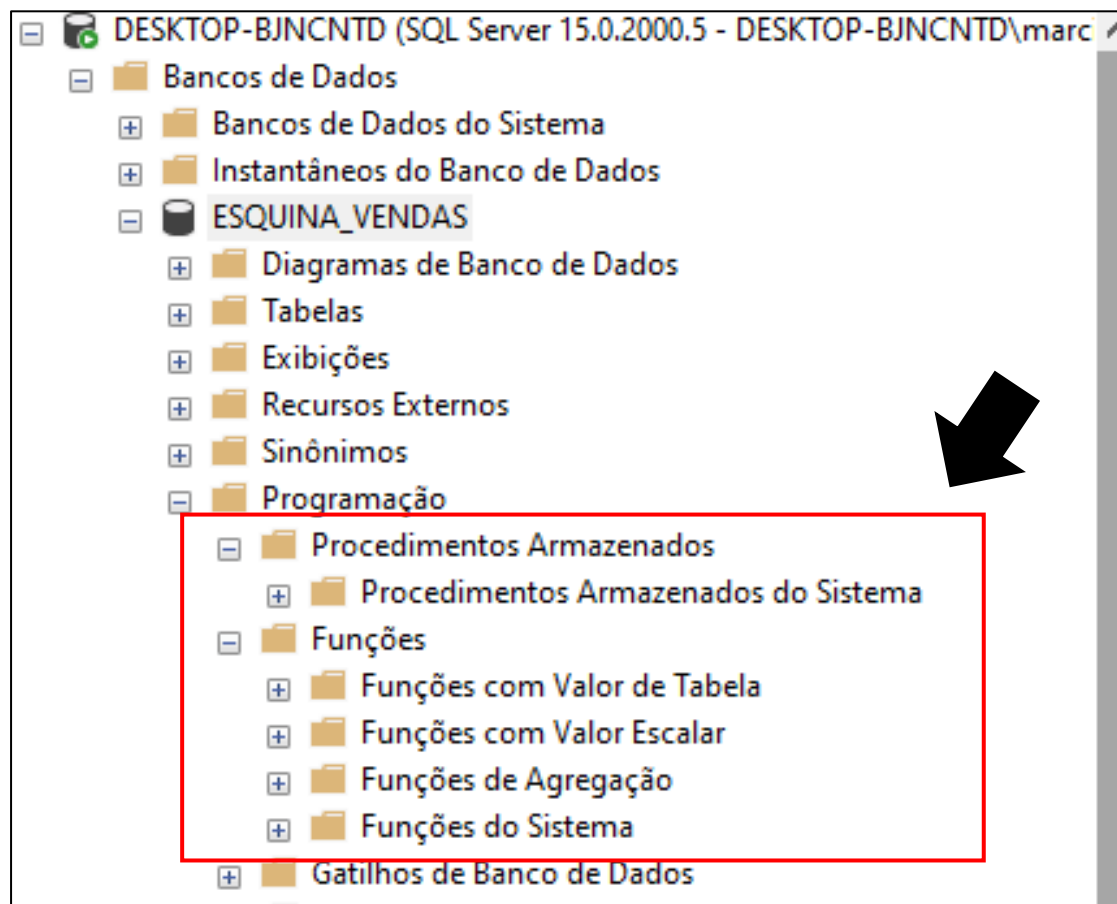
GO

T-SQL (Transact SQL)

```
DECLARE @CONTADOR INT
SET @CONTADOR = 1
WHILE (@CONTADOR <= 5)
BEGIN
    PRINT @CONTADOR
    SET @CONTADOR = @CONTADOR + 1
END
GO
```


T-SQL (Transact SQL)

- Criar Functions e Store Procedures



T-SQL (Transact SQL)

Esta mesma abordagem de possuir uma linguagem proprietária também se aplica a outros banco de dados.

- Oracle: PL/SQL
- IBM: SQL PL
- Sybase: T-SQL

Obs.: Sybase e SQL Server tiveram a mesma origem. Atualmente o nome da linguagem é a mesma, porém o seu conteúdo é muito diferente entre ambas.

Declarando variáveis no T-SQL

DECLARE <Nome da variável> <Tipo da variável>

- O nome deve ser único e iniciado com @
- O tipo da variável pode ser qualquer um existente no SQL Server, inclusive podemos ter variáveis do tipo tabelas (Veremos mais a frente este tipo especial)
- O valor da variável pode ser atribuída pelo comando SET ...
- ... ou através do comando SELECT

DESAFIO



Desafio 01

Crie 4 variáveis com as características abaixo:

- Nome: Cliente. Tipo: Caracteres com 10 posições. Valor: João
- Nome: Idade. Tipo: Inteiro. Valor: 10
- Nome: DataNascimento. Tipo: Data. Valor: 10/01/2007
- Nome: Custo. Tipo: Número com casas decimais. Valor: 10,23

Construa um script que declare estas variáveis, atribua valores a elas e exiba-as na saída do SQL Server Management Studio.

DESAFIO



Desafio 02

Crie uma variável chamada NUMNOTAS e atribua a ela o número de notas fiscais do dia 01/01/2017. Mostre na saída do script o valor da variável.

Controle de Fluxo

IF Expressão booleana

{Instruções em SQL}

[ELSE

{Instruções em SQL}]

- Expressão booleana → É uma expressão que retorna **TRUE** ou **FALSE**. Se a expressão lógica contiver uma instrução **SELECT**, a instrução **SELECT** deverá ser incluída entre parênteses.
- Instruções em SQL → É qualquer instrução ou agrupamento de instruções em **T-SQL**, conforme definido dentro do **IF** ou **ELSE**.

DESAFIO



Desafio 03

Crie um script que, baseado em uma data, conte o número de notas fiscais. Se houver mais de 70 notas, exiba a mensagem "Muita nota". Se não, exiba a mensagem "Pouca nota". Exiba também o número de notas.

DESAFIO



Desafio 04

Baseado no script de resposta do exercício anterior:

Em vez de testar com a variável @NUMNOTAS, use a própria consulta SQL na condição de teste.

Looping

WHILE Expressão booleana
{Instruções em SQL}

- Expressão booleana → É uma expressão que retorna **TRUE** ou **FALSE**. Se a expressão contiver uma instrução **SELECT**, a esta deverá ser incluída entre parênteses.
- Instruções em SQL → Qualquer instrução em **T-SQL** ou agrupamento de instruções em **SQL**
- **BREAK** → Provoca uma saída do loop **WHILE** mais interno. Todas as instruções que apareçam depois da palavra-chave **END**, que marca o final do loop, serão executadas.
- **CONTINUE** → Faz com que o loop **WHILE** seja reiniciado, ignorando todas as instruções depois da palavra **CONTINUE**.

DESAFIO



Desafio 05

Sabendo que a função abaixo soma um dia a uma data:

```
SELECT DATEADD(DAY, 1, @DATA)
```

Faça um script que, a partir do dia 01/01/2017, conte o número de notas fiscais até o dia 10/01/2017. Imprima a data e o número de notas fiscais.

Dicas:

- Declare variáveis do tipo DATE: DATAINICIAL e DATAFINAL;
- Faça um loop testando se a data inicial é menor que a data final;
- Imprima a data e o número de notas na saída do Management Studio. Não esqueça de converter as variáveis para VARCHAR;
- Acrescente um dia à data.

DESAFIO



Desafio 06

Continue evoluindo o script da resposta do exercício anterior. Agora, inclua o dia e o número de notas em uma tabela.

Tabelas temporárias

Temos 3 tipos de tabelas temporárias:

- Seus nomes começam com # → Tabelas que valem para a conexão vigente.
- Seus nomes começam com ## → Tabelas que valem para várias conexões.
- Seus nomes começam com @ → Tabelas que valem para o procedimento que está sendo executado.

Todas as tabelas são criadas em memória.

O que aprendemos nesta aula

- Como atribuir valores a estas variáveis;

- O uso de desvios de fluxo (IF);

- Como usar o loop;

- Algumas estruturas principais do T-SQL, como a declaração de variáveis;



- As tabelas temporárias;