



Developing Database

01 – T-SQL: Internals, Lotes, Operadores e Variáveis

Sand Onofre

Sand.Onofre@FaculdadeImpacta.com.br

Sumário

- Introdução ao Transact-SQL
 - Data Types Internals e Functions: Datetime x Smalldatetime
 - Expressões, Operadores e Categorias
 - Declaração de Variáveis
 - Diretivas de Batches
 - Exemplos

- Exercícios

DATA TYPES INTERNALS: SMALLDATETIME

- Armazenamento em 4 Bytes
- Armazenamento de Data e/ou Hora
- Precisão de MINUTOS (aproximação de segundos, final 00)
- Domínio de Data: 1º de janeiro de 1900 a 6 de junho de 2079
- Domínio de Hora: 00:00:00 a 23:59:00
- Valor Padrão (default): 1900-01-01 00:00:00

DATA TYPES INTERNALS: DATETIME

- Armazenamento em 8 Bytes
- Armazenamento de Data e/ou Hora
- Precisão aproximada, 3 MILISSEGUNDOS (aproximação dos milissegundos, finais 0, 3, 7)
- Domínio de Data: Janeiro 1, 1753, a dezembro 31, 9999
- Domínio de Hora: 00:00:00.000 a 23:59:59.997
- Valor Padrão (default): 1900-01-01 00:00:00.000

DATA TIME FUNCTIONS

- GETDATE(), YEAR(date), MONTH(date), DAY(date), ISDATE(date)
- DATEADD (datapart, number, date)
- DATEDIFF (datapart, startdate, enddate)
- DATEPART (datepart, date)

<i>datepart</i>	Abreviações
year	yy, yyyy
quarter	qq, q
month	mm, m
dayofyear	dy, y
day	dd, d
week	wk, ww

<i>datepart</i>	Abreviações
hour	hh
minute	mi, n
second	ss, s
millisecond	ms
microsecond	mcs
nanosecond	ns

Demonstração de FUNÇÕES APLICADAS

- GETDATE
- YEAR, MONTH, DAY
- ISDATE
- CAST
- CONVERT
- DATEADD
- DATEDIFF
- DATEPART

Exercício

- Faça um SELECT cujo algoritmo traga o 1o. e último dia do mês atual ?
- Faça um SELECT que retorne apenas a data de ontem, sem hora, minuto, segundo ou milissegundo.
- Faça um SELECT que retorne a data de ontem juntamente com a última hora, minuto, segundo e milissegundo.

T-SQL: Variáveis

Em T-SQL, uma variável local é um objeto que pode armazenar um simples valor de dado de um tipo específico

Variáveis em blocos e scripts são tipicamente usadas para:

Contar o número de vezes que um loop está sendo executado

Manter um dado para ser testado por uma declaração de controle de fluxo

Salvar valores de dados retornados por determinadas funções

```
DECLARE @local_variable <data_type>
```

```
declare @food varchar(20)
set @food = 'ice cream'
...
WHERE Description = @food
```



T-SQL: Categorias de Operadores

Type	Operators
Operadores Aritméticos	+, -, *, /, % <i>Vacation + SickLeave AS 'Total PTO'</i>
Operadores de Atribuição	=, +=, -=, *=, /= <i>SET @MyCounter = 1</i>
Operadores de Comparação	=, <, >, <>, !=, >=, <= <i>IF (@MyProduct <> 0) ...</i>
Operadores Lógicos	AND, OR, NOT <i>WHERE Department = 'Sales' AND (Shift = 'Evening' OR Shift = 'Night')</i>
Operador de Concatenação de Strings	+ <i>SELECT LastName + ', ' + FirstName AS Moniker</i>

T-SQL: Expressões e Operadores

- Podem ser utilizados para atribuição de valores

```
SET @DELTA = 56
```

- Essenciais para aplicação de cálculos

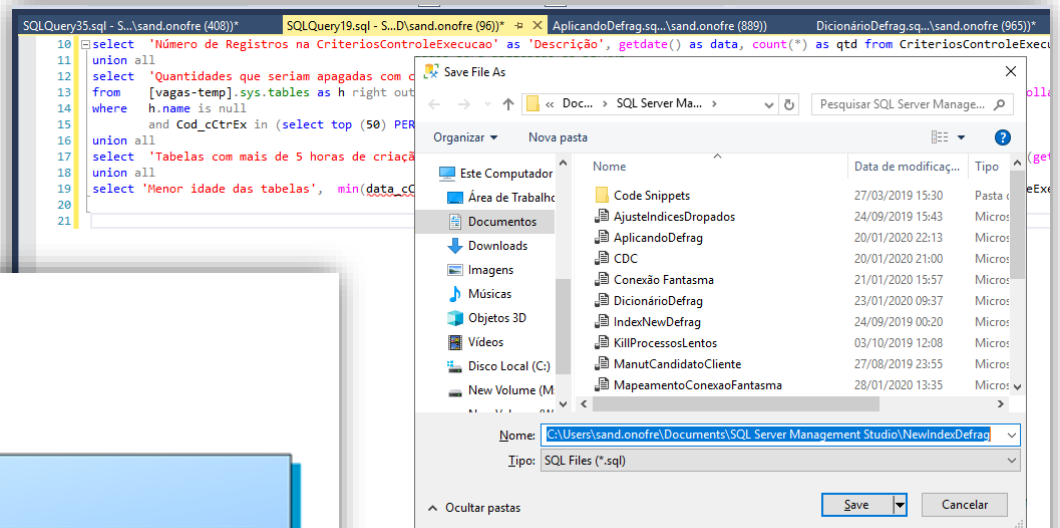
```
SET @DELTA = @DELTA + 100
```

```
SET @DELTA += 100
```

- Também servem para avaliar uma ou mais expressões que retornam os valores possíveis: TRUE, FALSE ou UNKNOWN.

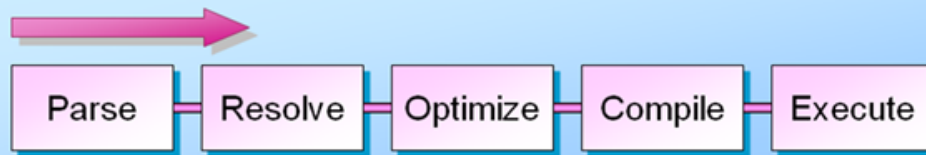
```
WHERE qtd = 100
```

T-SQL: Queries, Scripts e Process



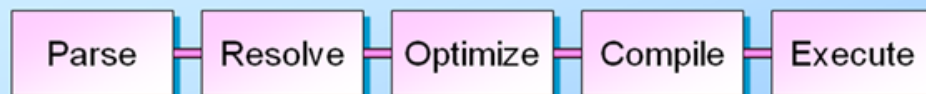
◆ How Queries Are Processed

Uncached Queries (Ad Hoc)



Cached Queries

First Execution



T-SQL: Diretivas de Batches (lotes)

- Lotes T-SQL são coleções de declarações, possuindo uma ou mais declarações T-SQL enviados para o SQL Server como uma unidade para análise, otimização e execução
- Os lotes são terminados por padrão com a declaração: GO
- Os lotes são limites para o escopo de variáveis
- Algumas declarações (por exemplo, CREATE FUNCTION, CREATE PROCEDURE, CREATE VIEW) não podem ser combinadas com outras no mesmo lote

```
CREATE VIEW <view_name>
AS ...;
GO
```

```
CREATE PROCEDURE <procedure_name>
AS ...;
GO
```

T-SQL: Diretivas de Batches (lotes)

- Os lotes são analisados para a sintaxe como uma unidade
- Os erros de sintaxe causam a rejeição de todo o lote

--Valid batch

```
INSERT INTO dbo.t1 VALUES(1,2,N'abc');
INSERT INTO dbo.t1 VALUES(2,3,N'def');
GO
```

--Invalid batch

```
INSERT INTO dbo.t1 VALUE(1,2,N'abc');
INSERT INTO dbo.t1 VALUES(2,3,N'def');
GO
```

T-SQL: Variáveis

- As variáveis são objetos que permitem o armazenamento de um valor para utilização mais tarde no mesmo **LOTE/BATCH**
- As variáveis são definidas com a palavra-chave DECLARE
- No SQL Server 2008 em diante, as variáveis podem ser declaradas e inicializadas na mesma declaração
- As variáveis são sempre locais para o lote em que são declaradas e ficam no escopo de seu lote

--Declare and initialize variables

```
DECLARE @numrows INT = 3, @catid INT, @qtd INT, @var INT;
SET @catid = 5;
SELECT @var = 2, @qtd = 1;
```

TSQL: Combinação de Operadores

Símbolos e operadores são avaliados para se obter um único valor

Expressões podem ser combinadas, se uma das afirmações abaixo for verdadeira:

As expressões possuem o mesmo tipo de dado

O tipo de dado com a prioridade mais baixa pode ser convertido para o tipo de dado com a mais elevada prioridade

SELECT @variavel + 2

T-SQL: Variáveis e Operadores

❑ Declarando Variáveis I

```
Declare @contador int
Select @contador
go
```

```
Declare @qtd tinyint, @texto varchar(20)
Set @qtd = 5
Set @texto = 'Aula'
Select @qtd as Quantidade, @texto Aula, @texto 'Aula de BD'
Go
```

```
Declare @qtd tinyint, @texto varchar(20)
Select @qtd = 150, @texto = 'Aula 2'
Select @qtd, @texto
go
```


T-SQL: Variáveis e Operadores

❑ Declarando Variáveis II

```
Declare @contador int = 5
Set @contador = @contador + 2
Select @contador
go
```

```
Declare @contador1 smallint = 15, @contador2 tinyint = 100
Set @contador1 = @contador1 + 5
Set @contador2 = @contador2 - 50
Select @contador1, @contador2, @contador1 * @contador2
go
```

```
Declare @contador1 smallint = 15, @contador2 tinyint = 100
Select @contador1 = @contador1 + 5, @contador2 = @contador2 - 50
Select @contador1, @contador2, @contador1 * @contador2
go
```

T-SQL: Variáveis e Operadores

❑ Declarando Variáveis III

```
Declare @contador int = 5
Set @contador += 2
Select @contador
go
```

```
Declare @contador1 smallint = 15, @contador2 tinyint = 100
Set @contador1 += 5
Set @contador2 -= 50
Select @contador1 * @contador2
go
```

```
Declare @qtd tinyint = 2, @texto varchar(20) = 'Aula'
Select @qtd =* 20, @texto = 'Aula' + '2'
Select @qtd , @texto
go
```

Demonstração de FUNÇÕES APLICADAS

- LEFT
- RIGHT
- SUBSTRING
- CHARINDEX
- REPLACE
- REVERSE
- LEN
- CONCAT

T-SQL: Exercícios com Operadores

1. Desenvolva T-SQL que resolva com precisão de números decimais os cálculos abaixo:

a. $\frac{456}{5}$ b. $782,23 \times 82,07$ c. $5^2 + 3^3$ d. $\sqrt{125}$

e. $((42 + 18) \div (54 \div 9) - 2) \div 4 - 2$

2. Através de funções built-in ou T-SQL, realize as operações abaixo:

- a. “Maria” e 5 espaços e “Antônia”
- b. Trocar a letra “a” por “e” na palavra “Tarara”
- c. Recuperar as 3 primeiras letras do texto “Mongaguá”
- d. Recuperar do caracter 6 ao 9 da palavra “Paralelepípedo”
- e. Recuperar as 2 últimas letras de “Inverno”

T-SQL: Exercícios com Operadores

- Com a tabela DEVDB criada com os nomes de alunos, gere um ÚNICO SELECT que retorne apenas o Nome e o Sobrenome de TODOS alunos. No **Result Set**, o Nome do aluno deverá ter a 1ª letra maiúscula, já o Sobrenome deve estar totalmente em maiúsculas. O nome da coluna deverá ser apelidada de **Nome e Sobrenome**.

nome		Nome e Sobrenome
abílio soares dos santos	→	Abílio SOARES
beatriz ribeiro		Beatriz RIBEIRO
JOSÉ ANDRADE SILVA		José SILVA
MARCELA J. ALVES		Marcela ALVES

T-SQL: Exercícios com Variáveis

4. Em T-SQL, declare as variáveis, obrigatoriamente com os menores tipos possíveis que comportem o valor inicial e final, execute alguma operação para obter o valor final solicitado.

Variável	Valor Inicial	Valor Final
Mes	5	12
Ano	2013	2100
graus	-10	32
id	5	60.000
placa	XPT-9563	XPT-9245
modelo	Cruze	Cruze LS
endereco	Av. Brasil	Av. Juscelino Kubitschek, 1000

T-SQL: Exercícios com Variáveis

5. Com os valores finais do exercício anterior e ainda em variáveis, faça as transformações solicitadas, se atentando a cálculos numéricos que devem ter o número de casas decimais apropriado:
 - a. $\text{Id} \div \text{Ano}$
 - b. Palavra “Mês” e 1 espaço e variável Mes
 - c. Ano e texto “ Mongaguá”
 - d. $(\text{id} \div \text{graus}) - (\text{Ano} * \text{Mês})$
 - e. Modelo, hífen e placa
 - f. Substituir o número atual no endereço e colocar os 4 últimos dígitos da placa

T-SQL: Exercícios com Variáveis

6. Você terá que deixar o script abaixo pronto para ser executado automaticamente através de um job. Faça as alterações suficientes para que seja executado de uma única vez (F5) e totalmente sem erros:

```
create table Teste_<iniciais do nome do aluno>
(
id tinyint primary key, nome varchar(20)
);
```

```
insert Teste_<iniciais do nome do aluno> values (1, 'Ana'), (2, 'Marcelo'), (3, 'Fábio'), (4, 'Paula');
```

```
create view vwTeste_<iniciais do nome do aluno>
as
select nome from Teste_<iniciais do nome do aluno>;
```

```
select * from vwTeste_<iniciais do nome do aluno>;
```




Obrigado !

Sand Onofre
Sand.Onofre@FaculdadeImpacta.com.br