

# Transact-SQL

Prof. André Luiz Montevecchi

[andreluiz@unipacbomdespacho.com.br](mailto:andreluiz@unipacbomdespacho.com.br)



**UNIPAC**  
*Campus BOM DESPACHO*

# T-SQL

- O Transact-SQL é a implementação do padrão International Standards Organization (ISO) ANSI-SQL de nível de entrada no SQL Server.

# T-SQL

- **Identificadores padrão:**
  - Um identificador que inicie com o símbolo de arroba (@) indica um parâmetro ou variável local.
  - Um identificador que inicie com um sinal de tralha (#) indica um procedimento ou uma tabela temporária.

# T-SQL

- **Identificadores delimitadores:**
  - Usados quando os nomes contiverem espaços incorporados.
  - Quando forem utilizadas palavras reservadas como nomes de objetos ou partes de nomes de objetos.



# T-SQL

- Os identificadores delimitados devem ser colocados entre colchetes ou aspas quando forem usados em instruções Transact-SQL.
  - `SELECT * FROM [ESPACO NA TABELA]`
  - `SELECT * FROM "ESPACO NA TABELA"`

# Variáveis locais

- Definidas pelo usuário com a instrução DECLARE.
- Recebem valores através de uma instrução SET ou SELECT.

```
- USE ODONTO
- DECLARE @NOME_DENTISTA VARCHAR(50),
          @NOME_CONVENIO VARCHAR(50)

SET @NOME_CONVENIO = 'UNIMED'
- SELECT @NOME_DENTISTA = DENTISTA.NOM_DENTISTA FROM DENTISTA
INNER JOIN CONVENIO ON
|CONVENIO.COD_CONVENIO = DENTISTA.COD_CONVENIO_DENTISTA
- WHERE CONVENIO.NOM_CONVENIO = @NOME_CONVENIO
- PRINT 'DENTISTA DO CONVENIO: ' + @NOME_DENTISTA
```

# Operadores

- Tipos de Operadores:
  - Aritmético
  - Comparação
  - Concatenação de seqüências de caracteres
  - Lógico



# Operadores aritméticos

- Os operadores aritméticos efetuam cálculos com colunas ou constantes numéricas.
- O Transact-SQL oferece suporte para operadores multiplicativos, incluindo multiplicação (\*), divisão (/) e módulo (%) e operadores aditivos, de adição (+) e subtração (-).



# Operadores de comparação

- Comparam duas expressões. É possível efetuar comparações entre variáveis, colunas e expressões semelhantes.

Operador	Significado
=	Igual a
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a
≠	Diferente de

# Operadores de concatenação

- O operador de concatenação de seqüências de caracteres (+) concatena valores de seqüência.

```
PRINT 'DENTISTA DO CONVENIO: ' + @NOME_DENTISTA
```

# Operadores Lógicos

- Os operadores lógicos AND, OR e NOT conectam condições de pesquisa em cláusulas WHERE.



# Controle de Fluxo

- Nível de instrução
  - Blocos BEGIN ... END
  - Blocos IF ... ELSE
  - Construções com WHILE
- Nível de registro
  - Expressão CASE



# Nível de Instrução

```
IF EXISTS (SELECT COD_CONVENIO FROM CONVENIO
            INNER JOIN DENTISTA ON
            DENTISTA.COD_CONVENIO_DENTISTA = CONVENIO.COD_CONVENIO
            WHERE NOM_DENTISTA = 'OTAVIO MACIEL')
    PRINT '*** DENTISTA NÃO PODE SER EXCLUÍDO! ***'
ELSE
    BEGIN
        DELETE FROM DENTISTA WHERE NOM_DENTISTA = 'OTAVIO MACIEL'
        PRINT '*** DENTISTA FOI EXCLUÍDO! ***'
    END
```



# Nível de Registro

- Sintaxe:

CASE *expressão*

    {WHEN *expressão THEN resultado*} [,...n]

    [ELSE *resultado*]

END



# Nível de Registro

```
SELECT VAL_PAGAMENTO AS 'VALOR', 'STATUS' =  
CASE  
WHEN (DAT_PAGAMENTO IS NULL)  
THEN 'NÃO PAGO!'  
ELSE 'PAGO'  
END  
FROM PAGAMENTO
```

	VALOR	STATUS
1	800.00	PAGO
2	6000.00	PAGO
3	2000.00	PAGO
4	3000.00	PAGO
5	2000.00	PAGO
6	7500.00	PAGO
7	2000.00	NÃO PAGO!

# Usando Transações

- As transações são grupos de instruções submetidas em conjunto. Contudo, o SGBD trata as transações como uma unidade de trabalho individual, e a transação pode ser bem-sucedida ou falhar como um todo. Esse processo mantém a integridade dos dados.

# Usando Transações

- As transações iniciam com uma instrução **BEGIN TRANSACTION** e terminam com uma instrução **COMMIT TRANSACTION** ou **ROLLBACK TRANSACTION**.



# Usando Transações

```
BEGIN TRANSACTION
DECLARE @ERRO BIT
SET @ERRO = 0
UPDATE CONTA_POUPANCA SET VALOR = (QUANTIA - 100)
    WHERE CORRENTISTA_ID = 7811
IF @@ERROR <> 0
    BEGIN
        PRINT 'OCORREU UM ERRO NA CONTA POUPANÇA!'
        SET @ERRO = 1
    END
UPDATE CONTA_CORRENTE SET VALOR = (QUANTIA + 100)
    WHERE CORRENTISTA_ID = 7811
IF @@ERROR <> 0
    BEGIN
        PRINT 'OCORREU UM ERRO NA CONTA CORRENTE!'
        SET @ERRO = 1
    END
IF @ERRO = 0
    COMMIT TRANSACTION
ELSE
    ROLLBACK TRANSACTION
```

