



# PROGRAMAÇÃO T-SQL

VIEWS, Procedures, triggers

PARA SER, UTP.





T-SQL

• É o cerne do SQL Server.

#### Variáveis

DECLARE @nome varchar(50) SET @nome = 'Jose' SELECT @nome

(No column name)
1 Jose

#### IF ... ELSE

DECLARE @nome varchar(50)
SET @nome = 'Jose'
IF @nome = 'Jose'
SELECT 'Nome eh '+@nome
ELSE
SELECT 'Nao conheco essa
pessoa.'

(No column name)

#### PARA SER, UTP.





# T-SQL

• É o cerne do SQL Server.

#### BEGIN...END – controle de fluxo

# DECLARE @nome varchar(50) SET @nome = 'Jose' IF @nome = 'Jose' BEGIN SELECT 'Aluno '+@nome SELECT '-----' SELECT 'Aprovado!' END ELSE SELECT 'Nao conheco essa pessoa.'

#### IF ... ELSE



#### PARA SER, UTP.



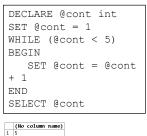
# PARA SER,

WHILE

T-SQL

• É o cerne do SQL Server.

#### BREAK



DECLARE @cont int
SET @cont = 1
WHILE (@cont < 5)
BEGIN
SET @cont = @cont
+ 1
IF @cont = 4
BREAK
END
SELECT @cont

CONTINUE

DECLARE @cont int

SET @cont = 1
WHILE (@cont < 5)
BEGIN
SET @cont = @cont
+ 1
CONTINUE
IF @cont = 4
BREAK
END
SELECT @cont







## **TRANSACTION**

uto adu br

- Transação é uma única unidade de trabalho.
  - Transações de confirmação automática : Cada instrução individual é uma transação.
  - Transações explícitas:
     Transação é iniciada com a instrução BEGIN
     TRANSACTION e finalizada com uma instrução COMMIT ou ROLLBACK.

```
BEGIN TRAN

UPDATE Planes SET

Number_of_seats =

Number_of_seats + 1

UPDATE Pilots SET

First_name = 'Sr/Sra' +

First_name

COMMIT TRAN
```

PARA SER, UTP.





## T-SQL

 Para desfazer as operações realizadas desde o início da transação.

#### ROLLBACK TRAN

```
BEGIN TRAN

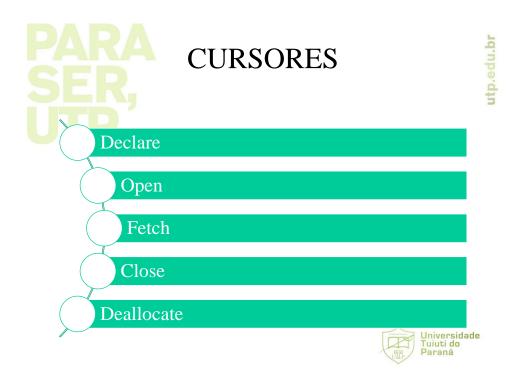
UPDATE Planes SET Number_of_seats = Number_of_seats + 1

UPDATE Pilots SET First_name = 'Sr/Sra' + First_name

IF @@ROWCOUNT = 5
COMMIT TRAN

ELSE
ROLLBACK TRAN
```





# PARA

## Cursores

DECLARE @PedidoID int, @Data date, @ItemID int, @ProdutoID int, @Descricao nvarchar(100), @Qtd real, @TOTAL real;

**DECLARE** cursor vendas **CURSOR FOR** 

SELECT p.PedidoID, p.Data, i.ItemID, i.ProdutoID,
pr.Descricao, i.Qtd, pr.Preco \* i.Qtd
FROM PEDIDOS AS p

INNER JOIN ITENS AS i

ON i.PedidoID = p.PedidoID

INNER JOIN PRODUTOS AS pr

ON pr.ProdutoID = i.ProdutoID;

PARA SER, UTP.



p.edu.br



## Cursores

OPEN cursor vendas;

FETCH NEXT FROM cursor\_vendas INTO @PedidoID, @Data, @ItemID, @ProdutoID, @Descricao, @Qtd, @TOTAL;

PRINT CONVERT(varchar, @PedidoID)+'
'+CONVERT(varchar, @Data)+' '+CONVERT(varchar,
@ItemID)+' '+CONVERT(varchar, @ProdutoID)+'
'+@Descricao+' '+CONVERT(varchar, @Qtd)+'
'+CONVERT(varchar, @TOTAL)

#### PARA SER, UTP.



# PARA

### Cursores

WHILE @@FETCH\_STATUS = 0 BEGIN

FETCH NEXT FROM cursor\_vendas INTO @PedidoID, @Data, @ItemID, @ProdutoID, @Descricao, @Qtd, @TOTAL;

PRINT CONVERT(varchar, @PedidoID)+' '+CONVERT(varchar, @Data)+' '+CONVERT(varchar, @ItemID)+' '+CONVERT(varchar, @ProdutoID)+' '+@Descricao+' '+CONVERT(varchar, @Qtd)+' '+CONVERT(varchar, @TOTAL)
END

CLOSE cursor vendas;

**DEALLOCATE** cursor vendas;

PARA SER, UTP.



ıtp.edu.br