



BANCO DE DADOS

UDF

*INLINE E
MULTI-STATEMENT*

**Prof. Fabiano Papaiz
IFRN**

UDF – FUNÇÕES DEFINIDAS PELO USUÁRIO

- Relembrando:
- *Inline Table-Valued*:
 - Retorna um conjunto de linhas e colunas, gerado a partir de um único comando *SELECT* em tabelas ou *views* do BD
 - Similar a utilização de *views*, mas com a diferença de que as funções podem receber parâmetros de entrada
- *Multi-statement Table-valued*:
 - Define explicitamente a estrutura da tabela que será retornada, com seus nomes de colunas e seus tipos de dados
 - Utilizada quando não podemos gerar o retorno a partir de um único *SELECT*, sendo necessário executar 2 ou mais *SELECTs* em sequência para gerar os dados, incluindo estes resultados em uma tabela temporária



EXEMPLOS

UDF – FUNÇÕES DEFINIDAS PELO USUÁRIO

- **EXEMPLO-1:** função *Inline* para retornar todas compras realizadas por um determinado cliente (banco *Northwind*):

```
create function obterVendasDoCliente(@cod_cliente varchar(5))
returns table
as
return(
    SELECT cli.CustomerID as Codigo_Cliente
           ,cli.CompanyName as Cliente
           ,emp.FirstName + ' ' + emp.LastName as Vendedor
           ,ven.OrderID as Codigo_Venda
           ,ven.OrderDate as Data_Venda
           ,ven.ShipCountry as Pais_Venda

    FROM Customers cli
    INNER JOIN Orders ven ON cli.CustomerID = ven.CustomerID
    inner join Employees emp on ven.EmployeeID = emp.EmployeeID
    where cli.CustomerID = @cod_cliente
)
```

UDF – FUNÇÕES DEFINIDAS PELO USUÁRIO

- **EXEMPLO-1:** função *Inline* para retornar todas compras realizadas por um determinado cliente (banco *Northwind*) :

```
create function obterVendasDoCliente(@cod_cliente varchar(5))
returns table
as
return(
    SELECT cli.CustomerID
           ,cli.CompanyName
           ,emp.FirstName
           ,ven.OrderDate
           ,ven.OrderID
           ,ven.ShipDate
    FROM Customers cli
    INNER JOIN Orders
    inner join Employees emp on ven.EmployeeID = emp.EmployeeID
    where cli.CustomerID = @cod_cliente
)
```

OBSERVAÇÕES:

- 1) O tipo de retorno (returns) deve ser **TABLE**
- 2) Não colocar o **BEGIN...END** no corpo da função (depois do **AS**)
- 3) Deve haver apenas a instrução **RETURN** no corpo da função, contendo apenas **1 único SELECT** entre parêntesis

UDF – FUNÇÕES DEFINIDAS PELO USUÁRIO

- Testando a função criada:

```
select *  
from dbo.obterVendasDoCliente('VINET')
```



	Codigo_Cliente	Cliente	Vendedor	Codig
1	VINET	Vins et alcools Chevalier	Steven Buchanan	1024
2	VINET	Vins et alcools Chevalier	Michael Suyama	1027
3	VINET	Vins et alcools Chevalier	Andrew Fuller	1029
4	VINET	Vins et alcools Chevalier	Andrew Fuller	1073
5	VINET	Vins et alcools Chevalier	Janet Leverling	1073

UDF – FUNÇÕES DEFINIDAS PELO USUÁRIO

○ *Funções Inline* X *Views*

- A vantagem de se criar uma função *Inline* ao invés de apenas utilizar uma *view*, é que com a função podemos assegurar que sempre serão passados parâmetros para sua execução
- Tais parâmetros geralmente são utilizados para filtrar o retorno do SELECT dentro da função
- Com isso, podemos evitar que uma *view* que retorne uma grande quantidade de registros (1 milhão por exemplo) seja executada sem que sejam especificados parâmetros de filtragem, sobrecarregando assim o SGBD

UDF – FUNÇÕES DEFINIDAS PELO USUÁRIO

- **EXEMPLO-2:** função *Multi-Statement* para retornar a quantidade de registros de algumas tabelas:

```
create function obterQuantidadeRegistros()
```

```
returns @TabelaRetorno table(  
    nome_tabela varchar(50)  
    ,qtd_registros int)
```

```
as
```

```
begin
```

```
--tabela de Clientes
```

```
insert into @TabelaRetorno  
select 'Clientes', count(*) from Customers
```

```
--tabela de Fornecedores
```

```
insert into @TabelaRetorno  
select 'Fornecedores', count(*) from Suppliers
```

```
--tabela de Vendas
```

```
insert into @TabelaRetorno  
select 'Vendas', count(*) from Orders
```

```
--no final devemos apenas chamar RETURN
```

```
return
```

```
end
```


UDF – FUNÇÕES DEFINIDAS PELO USUÁRIO

- **EXEMPLO-2:** função *Multi-Statement* para retornar a quantidade de registros de algumas tabelas:

```
create function obterQuantidadeRegistros()
```

```
returns @TabelaRetorno table(  
    nome_tabela varchar(50)  
    ,qtd_registros int)
```

```
as
```

```
begin
```

```
--tabela de Clientes
```

```
insert into @TabelaRetorno
```

```
select 'Clientes', count(*) from Customers
```

```
--tabela de Fornecedores
```

```
insert into @TabelaRetorno
```

```
select 'Fornecedores', count(*) from Suppliers
```

```
--tabela de Vendas
```

```
insert into @TabelaRetorno
```

```
select 'Vendas', count(*) from Sales
```

```
--no final devemos retornar
```

```
return
```

```
end
```

OBSERVAÇÕES:

- 1) Devemos dar um nome para a tabela temporária que será retornada, no exemplo foi **@TabelaRetorno**
- 2) Devemos inserir os registros que serão retornados nesta tabela temporária

UDF – FUNÇÕES DEFINIDAS PELO USUÁRIO

- Testando a função criada:

```
select *  
from dbo.obterQuantidadeRegistros()
```



Resultados		Mensagens
	nome_tabela	qtd_registros
1	Cientes	91
2	Fornecedores	29
3	Vendas	830

UDF – FUNÇÕES DEFINIDAS PELO USUÁRIO

- **EXEMPLO-3:** função *Multi-Statement* para retornar as empresas com as quais temos algum relacionamento:

```
create function obterEmpresasComQuaisMeRelaciono()  
returns @TabelaRetorno table(  
    nome_empresa varchar(100)  
    ,cidade varchar(50)  
    ,pais varchar(50) )
```

```
as  
begin
```

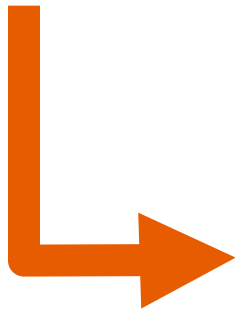
```
    --tabela de Clientes  
    insert into @TabelaRetorno  
    select companyName, city, country from Customers  
    --tabela de Fornecedores  
    insert into @TabelaRetorno  
    select companyName, city, country from Suppliers  
    --tabela de Expedidores  
    insert into @TabelaRetorno  
    select companyName, [null], [null], Orders from shippers  
    --no final devemos apenas chamar RETURN  
return  
end
```

Shippers não possui os campos City e Country

UDF – FUNÇÕES DEFINIDAS PELO USUÁRIO

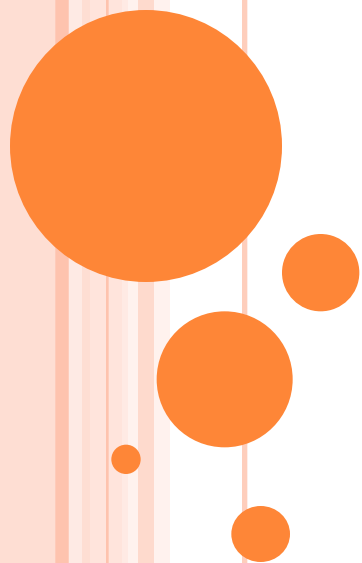
- Testando a função criada:

```
select *  
from dbo.obterEmpresasComQuaisMeRelaciono()  
order by nome_empresa
```



	nome_empresa	cidade	pais
22	Du monde entier	Nantes	France
23	Eastern Connection	London	UK
24	Ernst Handel	Graz	Austria
25	Escargots Nouveaux	Montceau	France
26	Exotic Liquids	London	UK
27	Familia Arquibaldo	Sao Paulo	Brazil
28	Federal Shipping	NULL	NULL

EXERCÍCIOS



UDF – FUNÇÕES DEFINIDAS PELO USUÁRIO

- **Exercício-1:**
- Em grupo do Projeto Integrador
- Com base no Projeto Integrador do seu grupo, analisem e especifiquem quais funções (UDF) poderiam ser criadas para auxiliar no desenvolvimento
 - Especificar qual seria o nome da função e o seu tipo de retorno (*Scalar*, *Inline* ou *Multi-Statement*)



FIM