Create Database OperadoresApply

Go

Use OperadoresApply

Go

Create Table Clientes

(CodigoCliente Int Identity(1,1) Primary Key Not Null,

NomeCliente VARCHAR(100) Not Null)

Go

Create Table Orcamento

(CodigoOrcamento Int Identity(1,1) Primary Key Not Null,

CodigoCliente Int Not Null,

Valor Decimal(10,2) Not Null,

Valor2 As (Valor/2),

Valor3 As (Valor/3),

Valor4 As (Valor/4))

Go

Insert Into Clientes (NomeCliente)

Values ('Pedro'),('Antonio'),('Galvão'),('Junior'),('Fernanda')

Go

Insert Into Orcamento(CodigoCliente, Valor)

VALUES (1,300), (1,400), (3,100),

(4,200), (1,1000), (1,700),

(3,300), (3,500), (4,700),

(4,5000)

Go

-- Exemplo 1 - Listar todos os clientes que tem orçamento e seus respectivos valores --

Select \* From Clientes Cli Inner Join Orcamento Orca

On Cli.CodigoCliente = Orca.CodigoCliente

Go

-- Cross Apply --

Select R.CodigoOrcamento, cli.NomeCliente,

R.Valor, R.Valor2, R.Valor3, R.Valor4

From Clientes Cli

Cross Apply (Select \* From Orcamento Orca

Where Cli.CodigoCliente = Orca.CodigoCliente) As R

Go

-- Exemplo 2 - Listar os dois maiores valores de orçamento de cada cliente --

Select \* From Clientes Cli

Cross Apply (Select Top 2 \* From Orcamento Orca

Where Cli.CodigoCliente = Orca.CodigoCliente

Order By Valor Desc) As R

Go

Select R.CodigoOrcamento, Cli.NomeCliente, R.Valor From Clientes Cli

Cross Apply (Select Top 2 \* From Orcamento Orca

Where Cli.CodigoCliente = Orca.CodigoCliente

Order By Valor Desc) As R

Go

-- Erro Inner Join não permite a referência externa

Select \* From Clientes Cli Inner Join (Select Top 2 \* From Orcamento Orca

Where Cli.CodigoCliente = Orca.CodigoCliente -- remover o Where

Order By Valor Desc ) B

ON Cli.CodigoCliente = Orca.CodigoCliente

Go

-- Exemplo 3 - Produto Cartesiano com Cross Apply --

Select Cli.NomeCliente, Orca.VALOR

From Clientes Cli Cross Apply Orcamento Orca

Go

-- Mesmo comportamento utilizando Cross Join --

Select Cli.NomeCliente, Orca.VALOR

From Clientes Cli Cross Join Orcamento Orca

Go

-- Exemplo 4 - Utilizando Função como filtro de Dados --

Select Cli.NomeCliente,

Orca.Valor,

R.ValorDividido

From Clientes Cli Inner Join Orcamento Orca

On Cli.CodigoCliente = Orca.CodigoCliente

Cross Apply FN\_DividirValor (Orca.Valor) As R

Go

-- Exemplo 5 - Utilizando Cross Apply para realizar Unpivot --

Select Cli.CodigoCliente,

U.ColunaUnpivot

From Clientes Cli INNER JOIN Orcamento Orca

ON Cli.CodigoCliente = Orca.CodigoCliente

Cross Apply (VALUES (Orca.VALOR),

(Orca.VALOR2),

(Orca.VALOR3),

(Orca.VALOR4)

) As U (ColunaUnpivot)

Go

-- Exemplo 6 - Listar todos os clientes que tem ou não orçamento e seus valores de orçamento --

Select \* From Clientes Cli

Outer Apply (Select \* From Orcamento Orca

Where Cli.CodigoCliente = Orca.CodigoCliente) As R

Go

-- Exemplo 7 - Listar a soma dos valores de todos os clientes que tem orçamento --

Select \* From Clientes Cli

Cross Apply (Select SUM(Valor) AS Somatoria From Orcamento Orca

Where Cli.CodigoCliente = Orca.CodigoCliente) As R

Go

select \* from orcamento

-- Teremos resultados sendo apresentados como nulo, para resolver vamos agrupar --

Select \* From Clientes Cli

Cross Apply (Select SUM(Valor) AS Somatoria,

AVG(Valor2) As MediaValor2

From Orcamento Orca

Where Cli.CodigoCliente = Orca.CodigoCliente

GROUP BY Orca.CodigoCliente) As R

Go

-- Utilizar a cláusula Where no Select principal em conjunto com o operador Is Not Null --

Select \* From Clientes Cli

Cross Apply (Select SUM(Valor) AS Somatoria From Orcamento Orca

Where Cli.CodigoCliente = Orca.CodigoCliente) As R

Where R.Somatoria Is Not Null

Go

-- Criando a Função --

Create FUNCTION FN\_DividirValor (@Valor Decimal(10,2))

Returns Table

AS

Return

(

Select @Valor/2 As ValorDividido

)

Go