



Faça como eu fiz

1) Antes de rodar comandos na prática no **SQL Server Management Studio**. Vá em **Ferramentas** opção **Opções**, **Resultados da Consulta**, **SQL Server**. Na caixa **Resultados em Grade** alteramos a opção **Dados em XML** de 2 MB para 5 MB.

2) No **SQL Server Management Studio** vemos se a base de dados **AdventureWorks2014** está selecionada.

3) Vamos fazer uma consulta de um cadastro de produtos. Entre com o código:

```
SELECT
Product.Name
,Product.ProductNumber as Number
,Product.ListPrice as Price
FROM Production.Product Product
WHERE Product.ListPrice > 0
```

```
AND Product.SellEndDate IS NULL  
FOR XML RAW;
```

[COPIAR CÓDIGO](#)

O retorno é uma tabela simples com várias informações. Apresenta um XML e um hyperlink que mostra o XML completo.

4) De maneira a melhorar a consulta vamos colocar as subcategorias dos produtos. Vamos fazer um **INNER JOIN** para incluir as subcategorias da tabela **Product**.

Fazemos estas ações com o seguinte código:

```
SELECT  
SubCategory.Name as SubCategoryName  
,Product.Name  
,Product.ProductNumber as Number  
,Product.ListPrice as Price  
FROM Production.Product Product  
INNER JOIN Production.ProductSubcategory  
Product.ProductSubcategoryID = SubCateg  
WHERE Product.ListPrice > 0
```

```
AND Product.SellEndDate IS NULL  
FOR XML RAW;
```

[COPIAR CÓDIGO](#)

O resultado do **FOR XML RAW** é uma tag única para cada linha.

5) Colocando mais um nível obtemos o seguinte código:

```
SELECT  
Category.Name as CategoryName  
,SubCategory.Name as SubCategoryName  
,Product.Name  
,Product.ProductNumber as Number  
,Product.ListPrice as Price  
FROM Production.Product Product  
INNER JOIN Production.ProductSubcategory  
Product.ProductSubcategoryID = SubCateg  
INNER JOIN Production.ProductCategory C  
Category.ProductCategoryID = SubCategor  
WHERE Product.ListPrice > 0  
AND Product.SellEndDate IS NULL  
FOR XML RAW;
```

[COPIAR CÓDIGO](#)

- 6) Vamos alterar os códigos anteriores para testar o **FOR XML AUTO**. O script fica desta maneira:

```
SELECT
Product.Name
,Product.ProductNumber as Number
,Product.ListPrice as Price
FROM Production.Product Product
WHERE Product.ListPrice > 0
AND Product.SellEndDate IS NULL
FOR XML AUTO;
```

[COPIAR CÓDIGO](#)

O resultado é uma pequena mudança. A tag tem o nome da tabela de onde vem dado. É uma tabela normalizada.

- 7) Alterando o segundo código para incluir o **FOR XML AUTO**.

```
SELECT
SubCategory.Name as SubCategoryName
```

```
,Product.Name  
,Product.ProductNumber as Number  
,Product.ListPrice as Price  
FROM Production.Product Product  
INNER JOIN Production.ProductSubcategory  
Product.ProductSubcategoryID = SubCateg  
WHERE Product.ListPrice > 0  
AND Product.SellEndDate IS NULL  
FOR XML AUTO;
```

[COPIAR CÓDIGO](#)

O resultado apresenta uma quebra no **XML**.
Separando em subcategorias. Foi criada uma
hierarquia.

8) O código seguinte fica assim:

```
SELECT  
Category.Name as CategoryName  
,SubCategory.Name as SubCategoryName  
,Product.Name  
,Product.ProductNumber as Number  
,Product.ListPrice as Price  
FROM Production.Product Product  
INNER JOIN Production.ProductSubcategory
```

```
Product.ProductSubcategoryID = SubCateg  
INNER JOIN Production.ProductCategory C  
Category.ProductCategoryID = SubCategor  
WHERE Product.ListPrice > 0  
AND Product.SellEndDate IS NULL  
FOR XML AUTO;
```

[COPIAR CÓDIGO](#)

O resultado é uma nova quebra de hierarquia.

9) Para retornar o resultado centrados em elementos incluir a palavra **ELEMENTS** no final da cláusula.

O código alterado é este:

```
SELECT  
Product.Name  
,Product.ProductNumber as Number  
,Product.ListPrice as Price  
FROM Production.Product Product  
WHERE Product.ListPrice > 0  
AND Product.SellEndDate IS NULL  
FOR XML AUTO, ELEMENTS;
```

[COPIAR CÓDIGO](#)

No segundo exemplo o código fica assim:

```
SELECT
SubCategory.Name as SubCategoryName
,Product.Name
,Product.ProductNumber as Number
,Product.ListPrice as Price
FROM Production.Product Product
INNER JOIN Production.ProductSubcategory
Product.ProductSubcategoryID = SubCateg
WHERE Product.ListPrice > 0
AND Product.SellEndDate IS NULL
FOR XML AUTO, ELEMENTS;
```

[COPIAR CÓDIGO](#)

O terceiro código fica desta maneira:

```
SELECT
Category.Name as CategoryName
,SubCategory.Name as SubCategoryName
```

```
,Product.Name  
,Product.ProductNumber as Number  
,Product.ListPrice as Price  
FROM Production.Product Product  
INNER JOIN Production.ProductSubcategory  
Product.ProductSubcategoryID = SubCategor  
INNER JOIN Production.ProductCategory C  
Category.ProductCategoryID = SubCategor  
WHERE Product.ListPrice > 0  
AND Product.SellEndDate IS NULL  
FOR XML AUTO, ELEMENTS;
```

[COPIAR CÓDIGO](#)