



Faça como eu fiz

- 1) No **SQL Server Management Studio**, crie uma nova consulta usando a base de dados **AdventureWorks2014**.
- 2) Vamos fazer uma consulta de um cadastro de produtos usando o código que escrevemos antes:

```
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, ELEMENTS;
```

[COPIAR CÓDIGO](#)

O código alterado fica assim:

```
SELECT  
Category.Name as "Caregory/CategoryName"  
,SubCategory.Name as "Caregory/SubCateg"  
,Product.ProductNumber as "Caregory/Sub"  
,Product.ListPrice as "Caregory/SubCate"  
,Product.Name as "Caregory/SubCategory/"  
FROM Production.Product Product  
INNER JOIN Production.ProductSubcategor  
Product.ProductSubcategoryID = SubCateg  
INNER JOIN Production.ProductCategory C  
Category.ProductCategoryID = SubCategor  
WHERE Product.ListPrice > 0  
AND Product.SellEndDate IS NULL  
FOR XML PATH, ELEMENTS;
```

[COPIAR CÓDIGO](#)

3) Vamos usar somente o SQL para fazer a escrita em um arquivo externo. Segue a Procedure mostrada no vídeo:

```
CREATE PROCEDURE dbo.usp_WriteXMLFile @  
@  
@  
  
AS  
  
BEGIN  
    SET NOCOUNT ON;  
    IF(OBJECT_ID('tempdb..##XML') I  
        DROP TABLE ##XML;  
    CREATE TABLE ##XML(XMLHolder XM  
    INSERT INTO ##XML(XMLHolder)  
        SELECT @XML;  
    DECLARE @cmd TABLE  
    (name          NVARCHAR(35),  
     minimum      INT,  
     maximum      INT,  
     config_value INT,  
     run_value     INT  
    );  
    DECLARE @run_value INT;  
    EXECUTE master.dbo.sp_configure  
    RECONFIGURE;  
    INSERT INTO @cmd (name, minimum
```

```
EXECUTE sp_configure 'xp_cmdshe
SELECT @run_value = run_value
FROM @cmd;
IF @run_value = 0
    BEGIN
        EXEC sp_configure
            'xp_cmdshell',
            1;
        RECONFIGURE;
    END;
DECLARE @SQL NVARCHAR(300)= '';
SET @SQL = 'bcp ##XML out "' +
EXECUTE master..xp_cmdshell @SQ
IF @run_value = 0
    BEGIN
        EXECUTE sp_configure 'x
        RECONFIGURE;
    END;
IF(OBJECT_ID('tempdb..##XML') I
    DROP TABLE ##XML;
SET NOCOUNT OFF;
END;
GO

DECLARE @x XML;
SET @x = (SELECT
```

```
Category.Name as "Caregory/CategoryName",
SubCategory.Name as "Caregory/SubCateg",
Product.ProductNumber as "Caregory/Sub",
Product.ListPrice as "Caregory/SubCate",
Product.Name as "Caregory/SubCategory/"
FROM Production.Product Product
INNER JOIN Production.ProductSubcategor
Product.ProductSubcategoryID = SubCateg
INNER JOIN Production.ProductCategory C
Category.ProductCategoryID = SubCategor
WHERE Product.ListPrice > 0
AND Product.SellEndDate IS NULL
FOR XML PATH, ELEMENTS)
```

```
exec usp_WriteXMLFile @x, 'C:\TEMP\Curs
```

[COPIAR CÓDIGO](#)

4) Como ler dados de um arquivo externo para trazer ao banco XML. O seguinte código faz o que é explicado no vídeo.

```
DECLARE @x XML;
SET @x = (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);
```

```
exec [dbo].[usp_WriteXMLFile] @x, 'C:\T
```

```
DECLARE @x XML;  
SET @x = (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);
```

```
exec [dbo].[usp_WriteXMLFile] @x, 'C:\T
```

```
CREATE PROCEDURE dbo.usp_LoadXMLFromFile  
AS  
BEGIN
```

```
SET NOCOUNT ON;
DECLARE @cmd TABLE
(
    name          NVARCHAR(35),
    minimum       INT,
    maximum       INT,
    config_value  INT,
    run_value     INT
);
DECLARE @run_value INT;
-- Save original configuration
INSERT INTO @cmd
(
    name,
    minimum,
    maximum,
    config_value,
    run_value
)
EXEC sp_configure
    'xp_cmdshell';
SELECT @run_value = run_value
FROM @cmd;
IF @run_value = 0
BEGIN
    EXEC sp_configure
        'xp_cmdshell',
        1;
    RECONFIGURE;
```

```
END;  
IF NOT EXISTS  
(  
    SELECT *  
    FROM sys.objects  
    WHERE object_id = OBJECT_ID  
        AND type IN(N'U')  
)  
    CREATE TABLE dbo._XML  
    (ID          INT NOT NULL  
    XMLFileName NVARCHAR(300),  
    XML_LOAD    XML,  
    Created     DATETIME  
    );  
ELSE  
    TRUNCATE TABLE dbo._XML;  
DECLARE @DOS NVARCHAR(300)= N''  
DECLARE @files TABLE  
(tID      INT IDENTITY(1, 1) NOT  
    XMLFile NVARCHAR(300)  
);  
SET @DirBaseLocation = IIF(RIGHT  
SET @DOS = 'dir /B /O:-D ' + @D  
INSERT INTO @files(XMLFile)  
EXEC master..xp_cmdshell  
    @DOS;  
IF @run_value = 0
```

```
BEGIN
    EXECUTE sp_configure
        'xp_cmdshell',
        0;
    RECONFIGURE;
END;
DECLARE cur CURSOR
FOR SELECT XMLFile
    FROM @files
    WHERE XMLFile LIKE '%.xml';
OPEN cur;
FETCH NEXT FROM cur INTO @FileN
WHILE @@FETCH_STATUS = 0
    BEGIN
        BEGIN TRY
            SET @SQL = 'INSERT
            EXECUTE sp_executes
            @SQL;
            FETCH NEXT FROM cur
        END TRY
        BEGIN CATCH
            SELECT @SQL,
                ERROR_MESSAGE
        END CATCH;
    END;
CLOSE cur;
DEALLOCATE cur;
```

```
SET NOCOUNT OFF;  
  
END;  
  
GO  
  
exec usp_LoadXMLFromFile 'C:\Temp\C  
  
SELECT * FROM _XML;
```

[COPIAR CÓDIGO](#)