

Faça como eu fiz

- 1) Para prosseguir a partir daqui é necessário que você tenha o **SQL Server 2017** ou superior e o **SQL Server Management Studio**. Caso você não tenha este ambiente volte na aula apontada pelo instrutor no vídeo e instale os produtos.
- 2) Baixe o arquivo **WideWorldImporters-Full.zip**, copie para um diretório local da sua máquina e descompacte-o.
- 3) Abra o **SQL Server Management Studio** e, com o botão da direita do mouse sobre **Banco de Dados** escolha a opção **Restaurar Banco de Dados**.
- 4) Clique na opção **Dispositivo**.
- 5) Selecione o arquivo que foi descompactado.
- 6) Clique em **OK**. O banco de dados será restaurado.
- 7) Execute a consulta:

```
SELECT TOP 2 [CustomerName]
           ,[PrimaryContact]
           ,[AlternateContact]
           ,[PhoneNumber]
FROM [WideWorldImporters].[Website].[Customers];
```

- 8) Teremos o resultado em forma de tabela. Porém se você fizer o mostrado abaixo:

```
SELECT TOP 2 [CustomerName]
           ,[PrimaryContact]
           ,[AlternateContact]
           ,[PhoneNumber]
FROM [WideWorldImporters].[Website].[Customers]
FOR JSON AUTO;
```

Iremos observar o resultado da consulta no formato JSON.

- 9) Para visualizar o resultado clique no HiperLink e a resposta será exibida no XML Editor. Copiando o texto e colando no JSON Editor OnLine podemos ver o resultado formatado.
- 10) A representação exibida através da cláusula FOR JSON AUTO trás a tabela no formato JSON usando uma lógica padrão: Um Array onde cada elemento representa uma linha da tabela e cada propriedade do JSON do Array, uma coluna.
- 11) Mas este formato depende de como foi construído o comando SQL. Se executarmos:

```
SELECT db_name() as [Database],
       sys.schemas.name,
       sys.objects.name,
       sys.columns.name FROM sys.objects
```

```

JOIN sys.schemas on sys.objects.schema_id = sys.schemas.schema_id
JOIN sys.columns ON sys.columns.object_id =
sys.objects.object_id JOIN ( SELECT TOP (1) o.object_id, count( c.name) [name]
FROM sys.columns c
JOIN sys.objects o ON c.object_id = o.object_id WHERE type = 'u'
GROUP BY o.object_id HAVING COUNT( c.name) < 6
)
countCol
ON countCol.object_id = sys.objects.object_id
WHERE type = 'u'
FOR JSON AUTO;

```

Veremos um JSON formatado de forma diferente. Isso vai depender dos JOINS que a nossa consulta estiver utilizando.

12) Veja agora esta nova consulta:

```

SELECT TOP 2 [CustomerName]
      ,[PrimaryContact]
      ,[AlternateContact]
      ,[PhoneNumber]
FROM [WideWorldImporters].[Website].[Customers]
WHERE [AlternateContact] IS NULL
FOR JSON AUTO;

```

Veja que a coluna **AlternateContact** não aparece nas propriedades do JSON porque seu valor é nulo para todas as linhas resultado da consulta SQL.

13) Se usarmos a cláusula **INCLUDE_NULL_VALUES** os nulos serão representados no JSON de saída.

```

SELECT TOP 2 [CustomerName]
      ,[PrimaryContact]
      ,[AlternateContact]
      ,[PhoneNumber]
FROM [WideWorldImporters].[Website].[Customers]
WHERE [AlternateContact] IS NULL
FOR JSON AUTO, INCLUDE_NULL_VALUES;

```

14) Note que, se você não usar a cláusula **INCLUDE_NULL_VALUES** numa consulta que, em algumas linhas retorna nulo e em outras não o resultado dos JSONs, dentro do Array, terão estruturas diferentes.

```

SELECT X.* FROM (
SELECT TOP 1 QUOTENAME([CustomerName]) AS CustomerName
      ,[PrimaryContact]
      ,[AlternateContact]
      ,[PhoneNumber]
FROM [WideWorldImporters].[Website].[Customers]
WHERE [AlternateContact] IS NOT NULL
UNION
SELECT TOP 1 [CustomerName]
      ,[PrimaryContact]
      ,[AlternateContact]
      ,[PhoneNumber]
FROM [WideWorldImporters].[Website].[Customers]

```

```
WHERE [AlternateContact] IS NULL) X
FOR JSON AUTO;
```

15) O retorno padrão do JSON, quando usamos a cláusula FOR JSON AUTO, apresenta um Array para representar o conjunto de linhas da saída da consulta. Muitas vezes precisamos omitir este "abre e fecha colchetes" do resultado da consulta. Para isso podemos usar a cláusula mostrada a seguir.

```
SELECT X.* FROM (
SELECT TOP 1 [CustomerName]
      ,[PrimaryContact]
      ,[AlternateContact]
      ,[PhoneNumber]
FROM [WideWorldImporters].[Website].[Customers]
WHERE [AlternateContact] IS NOT NULL
UNION
SELECT TOP 1 [CustomerName]
      ,[PrimaryContact]
      ,[AlternateContact]
      ,[PhoneNumber]
FROM [WideWorldImporters].[Website].[Customers]
WHERE [AlternateContact] IS NULL) X
FOR JSON AUTO, WITHOUT_ARRAY_WRAPPER;
```

16) Outra possibilidade é criar uma propriedade raiz no JSON de saída. Execute:

```
SELECT TOP 2 [CustomerName]
      ,[PrimaryContact]
      ,[AlternateContact]
      ,[PhoneNumber]
FROM [WideWorldImporters].[Website].[Customers]
FOR JSON AUTO, ROOT('Customer');
```

17) Vimos, ainda nesta aula, que, dependendo do número de JOINS efetuados na consulta o formato do JSON muda. Usando a cláusula FOR JSON AUTO a representação fica mais próxima de uma representação de um organograma quando hierarquizamos a saída dos dados.

```
SELECT TOP 3 [CustomerName]
      ,[PrimaryContact]
      ,[AlternateContact]
      ,[PhoneNumber]
FROM [WideWorldImporters].[Website].[Customers]
WHERE [AlternateContact] IS NOT NULL
FOR JSON AUTO;
```

18) Compare a consulta acima com a apresentada a seguir. Aqui usamos a cláusula FOR JSON PATH para apresentar esta hierarquização da saída em forma de tabela des-normalizada.

```
SELECT TOP 3 [CustomerName]
      ,[PrimaryContact]
      ,[AlternateContact]
      ,[PhoneNumber]
```

```
FROM [WideWorldImporters].[Website].[Customers]
WHERE [AlternateContact] IS NOT NULL
FOR JSON PATH;
```