

Faça como eu fiz

1) Podemos observar que na tabela **Sales.Invoices** temos um campo chamado **ReturnedDeliveryData** que é do tipo JSON. Já o campo **ConfirmedReceivedBy** não é do tipo JSON. Podemos identificar se o conteúdo do campo é ou não um JSON válido através da função ISJSON. Veja:

```
SELECT ConfirmedReceivedBy, ReturnedDeliveryData
,ISJSON(ConfirmedReceivedBy), ISJSON(ReturnedDeliveryData)
FROM SALES.Invoices;
```

Ele vai retornar 1 se o conteúdo é um JSON e 0 se não for.

2) Esta função pode ser usada para identificar se o JSON está no formato correto. veja abaixo:

```
SELECT ISJSON('{"NOME":"JOÃO"}')
```

3) O script abaixo permite que possa ser identificado, em todas as tabelas da base, quais os campos que são do tipo JSON.

```
SET NOCOUNT ON;
DECLARE @SQL nvarchar(1000)
IF (OBJECT_ID('tempdb.dbo.# Result')) IS NOT NULL
DROP TABLE #Result
CREATE TABLE #Result (tblName nvarchar( 200),
clmnName nvarchar( 100),
DateType nvarchar( 100),
JSONDoc nvarchar( MAX),)

DECLARE cur CURSOR
FOR SELECT 'SELECT TOP (1) ''' + QUOTENAME(s.name) + '.' + QUOTENAME(o.name) + ''' as TblName, '
+ ''' as ClmName, ''' + t.name + QUOTENAME( case c.max_length when -1 then 'MAX' ELSE
cast( c.max_length as varchar( 5)) END , ''') + ''' as DataType, ' + QUOTENAME( c.name)
+ ' FROM '
+ QUOTENAME( s.name) + '.' + QUOTENAME( o.name) +
' WHERE ISJSON(' + QUOTENAME( c.name) + ') = 1;'
FROM sys.columns c JOIN sys.types t on c.system_type_id = t.system_type_id
JOIN sys.objects o ON c.object_id = o.object_id AND o.type = 'u'
JOIN sys.schemas s ON s.schema_id = o.schema_id WHERE t.name IN(' varchar', 'nvarchar')
AND (c.max_length = -1 OR c.max_length > 100)

OPEN cur FETCH NEXT FROM cur INTO @SQL
WHILE @@FETCH_STATUS = 0
BEGIN
    print @SQL
    INSERT #Result
    EXEC(@SQL)
    FETCH NEXT FROM cur INTO @SQL
END
DEALLOCATE cur;
SELECT JSONDoc, tblName, clmnName, DateType FROM #Result
```

```
ORDER BY tblName, clmnName
DROP TABLE #Result
SET NOCOUNT OFF;
```

4) Para colocar o conhecimento em prática construa o seguinte script.

```
DECLARE @JSON VARCHAR(MAX);

SET @JSON = '{
  "CustomerName": "Tailspin Toys (Head Office)",
  "PrimaryContact": "Waldemar Fisar",
  "AlternateContact": "Laimonis Berzins",
  "PhoneNumber": [
    "(308) 555-0100",
    "(308) 222-4436"
  ],
  "Address": {
    "Street": "1St Canal",
    "Number": 23,
    "City": "Atlanta"
  },
  "Invoices": [
    {
      "InvoiceDate": "2013-03-14",
      "InvoiceValue": 22500,
      "InvoiceProducts": [
        {
          "NameProduct": "Fruit Juice",
          "Value": 12500
        },
        {
          "NameProduct": "Ice Tea",
          "Value": 10000
        }
      ]
    },
    {
      "InvoiceDate": "2014-04-12",
      "InvoiceValue": 20000,
      "InvoiceProducts": [
        {
          "NameProduct": "Soda",
          "Value": 15000
        },
        {
          "NameProduct": "Ice Tea",
          "Value": 5000
        }
      ]
    }
  ]
}'

SELECT @JSON;
```

A partir daqui as consultas apresentadas serão apenas descritas com o último SELECT. Mas, para executá-las a declaração da variável @JSON e sua atribuição deverão ser sempre executadas conforme mostrado no vídeo.

5) Dependendo do que eu queira extrair do JSON tenho que usar a referência a uma determinada propriedade através do \$, seguido de um ponto e a referência a propriedade. Se quero obter dados de uma propriedade que está no primeiro nível do JSON executamos:

```
SELECT JSON_VALUE(@JSON, '$.CustomerName')
```

Note que o valor da propriedade é extraída da variável @JSON.

6) Se queremos extrair um elemento de um Array usamos:

```
SELECT JSON_VALUE(@JSON, '$.PhoneNumber[0]')
```

7) Se desejamos extrair uma propriedade que está no meio do JSON (Alguns níveis abaixo) executamos:

```
SELECT JSON_VALUE(@JSON, '$.Invoices[0].InvoiceProducts[0].NameProduct')
```

8) Usando JSON_VALUE podemos exibir os dados do JSON como se fosse uma tabela. Para a consulta abaixo não é preciso declarar a variável #JSON e nem atribuir o valor a ela.

```
SELECT JSON_VALUE(ReturnedDeliveryData, '$.Events[0].Event') AS Event_0
, JSON_VALUE(ReturnedDeliveryData, '$.Events[0].EventTime') AS EventTime_0
, JSON_VALUE(ReturnedDeliveryData, '$.Events[0].ConNote') AS ConNote_0
, JSON_VALUE(ReturnedDeliveryData, '$.Events[1].Event') AS Event_1
, JSON_VALUE(ReturnedDeliveryData, '$.Events[1].EventTime') AS EventTime_1
, JSON_VALUE(ReturnedDeliveryData, '$.Events[1].ConNote') AS ConNote_1
, JSON_VALUE(ReturnedDeliveryData, '$.Events[1].DriverID') AS DriverID
, JSON_VALUE(ReturnedDeliveryData, '$.Events[1].Status') AS ConNote_1
FROM SALES.Invoices;
```

9) Inclusive usar a propriedade extraída como filtro.

```
SELECT JSON_VALUE(ReturnedDeliveryData, '$.Events[1].DriverID') AS DRIVERID, * FROM SALES.Invoices
WHERE JSON_VALUE(ReturnedDeliveryData, '$.Events[1].DriverID') = 17
```

10) Mas o JSON_VALUE somente extrai dados de propriedade que possuem valores simples (Números, textos, datas ou valores lógicos). Quando o valor de uma propriedade é um Array ou um outro JSON usamos uma função diferente. Execute:

```
SELECT JSON_QUERY(@JSON, '$.PhoneNumber')
```

Aqui temos que executar a consulta com a variável @JSON e sua atribuição de valor.

11) Execute a seguinte consulta:

```
SELECT JSON_VALUE(UserPreferences,'$.theme') AS theme
,JSON_VALUE(UserPreferences,'$.dateFormat') AS dateFormat
,JSON_VALUE(UserPreferences,'$.timeZone') AS timeZone
,JSON_VALUE(UserPreferences,'$.table.pagingType') AS pagingType
,JSON_VALUE(UserPreferences,'$.table.pageLength') AS pageLength
,JSON_VALUE(UserPreferences,'$.favoritesOnDashboard') AS favoriteOnDashboard
FROM Application.People;
```

Note que o retorno desta consulta mostra as propriedades do JSON contido no campo UserPreferences em forma de tabela.

12) Podemos repetir este processo de outra maneira:

```
SELECT
UserPref.theme
,UserPref.timeZone
,UserPref.pagingType
,UserPref.pageLength
,UserPref.favoriteOnDashboard
FROM Application.People
CROSS APPLY OPENJSON(UserPreferences)
WITH
(theme VARCHAR(20) '$.theme'
,dateFormat VARCHAR(20) '$.dateFormat'
,timeZone VARCHAR(20) '$.timeZone'
,pagingType VARCHAR(20) '$.table.pagingType'
,pageLength int '$.table.pageLength'
,favoriteOnDashboard bit '$.favoritesOnDashboard'
) AS UserPref;
```