# 31. Introducing XML

declare @x xml

set @x = '<Shopping ShopperName="Phillip Burton" Weather="Nice">

<ShoppingTrip ShoppingTripID="L1">

<Item Cost="5">Bananas</Item>

<Item Cost="4">Apples</Item>

<Item Cost="3">Cherries</Item>

</ShoppingTrip>

<ShoppingTrip ShoppingTripID="L2">

<Item>Emeralds</Item>

<Item>Diamonds</Item>

<Item>Furniture</Item>

</ShoppingTrip>

</Shopping>'

select @x

update [dbo].[tblEmployee]

set XMLOutput = @x

where EmployeeNumber = 200

select \* from [dbo].[tblEmployee]

# 32. FOR XML RAW

select E.EmployeeNumber, E.EmployeeFirstName, E.EmployeeLastName

, E.DateOfBirth, T.Amount, T.DateOfTransaction

from [dbo].[tblEmployee] as E

left join [dbo].[tblTransaction] as T

on E.EmployeeNumber = T.EmployeeNumber

where E.EmployeeNumber between 200 and 202

for xml raw('MyRow'), elements ( type of attribute and element for tag)

# 33. FOR XML AUTO

select E.EmployeeNumber, E.EmployeeFirstName, E.EmployeeLastName

, E.DateOfBirth, T.Amount, T.DateOfTransaction

from [dbo].[tblEmployee] as E

left join [dbo].[tblTransaction] as T

on E.EmployeeNumber = T.EmployeeNumber

where E.EmployeeNumber between 200 and 202

for xml auto, elements

# 34. FOR XML PATH

select E.EmployeeFirstName as '@EmployeeFirstName'

, E.EmployeeLastName as '@EmployeeLastName'

, E.EmployeeNumber

, E.DateOfBirth

, T.Amount as 'Transaction/Amount'

, T.DateOfTransaction as 'Transaction/DateOfTransaction#'

from [dbo].[tblEmployee] as E

left join [dbo].[tblTransaction] as T

on E.EmployeeNumber = T.EmployeeNumber

where E.EmployeeNumber between 200 and 202

for xml path('Employees'), ROOT('MyXML')

# 35. FOR XML EXPLICIT

select 1 as Tag, NULL as Parent

, E.EmployeeFirstName as [Elements!1!EmployeeFirstName]

, E.EmployeeLastName as [Elements!1!EmployeeLastName]

, E.EmployeeNumber as [Elements!1!EmployeeNumber]

, E.DateOfBirth as [Elements!1!DateOfBirth]

, null as [Elements!2!Amount]

, null as [Elements!2!DateOfTransaction]

from [dbo].[tblEmployee] as E

where E.EmployeeNumber between 200 and 202

union all

select 2 as Tag, 1 as Parent

, null as [EmployeeFirstName]

, null as [EmployeeLastName]

, T.EmployeeNumber

, null as DateOfBirth

, Amount

, DateOfTransaction

from [dbo].[tblTransaction] as T

inner join [dbo].[tblEmployee] as E on T.EmployeeNumber = E.EmployeeNumber

where T.EmployeeNumber between 200 and 202

order by EmployeeNumber, [Elements!2!Amount]

for xml explicit

# 35. XQuery Value method

declare @x xml

set @x='<Shopping ShopperName="Phillip Burton" >

<ShoppingTrip ShoppingTripID="L1" >

<Item Cost="5">Bananas</Item>

<Item Cost="4">Apples</Item>

<Item Cost="3">Cherries</Item>

</ShoppingTrip>

<ShoppingTrip ShoppingTripID="L2" >

<Item>Emeralds</Item>

<Item>Diamonds</Item>

<Item>Furniture</Item>

</ShoppingTrip>

</Shopping>'

select @x.value('(/Shopping/ShoppingTrip/Item/@Cost)[1]','varchar(50)')

# 36. XQuery Modify method

set @x.modify('replace value of (/Shopping/ShoppingTrip[1]/Item[3]/@Cost)[1]

with "6.0"')

select @x

set @x.modify('insert <Item Cost="5">New Food</Item>

into (/Shopping/ShoppingTrip)[2]')

select @x

# 37. XQuery Query and FLWOR 1

select @x.query('for $ValueRetrieved in /Shopping/ShoppingTrip/Item

return $ValueRetrieved')

select @x.query('for $ValueRetrieved in /Shopping/ShoppingTrip/Item

return string($ValueRetrieved)')

select @x.query('for $ValueRetrieved in /Shopping/ShoppingTrip[1]/Item

return concat(string($ValueRetrieved),";")')

# 38. XQuery Query and FLWOR 2

select @x.query('for $ValueRetrieved in /Shopping/ShoppingTrip[1]/Item

let $CostVariable := $ValueRetrieved/@Cost

where $CostVariable >= 4

order by $CostVariable

return concat(string($ValueRetrieved),";")')

# 39. nodes using Variable (shredding a variable)

select tbl.col.value('.', 'varchar(50)') as Item

, tbl.col.value('@Cost','varchar(50)') as Cost

into tblTemp

from @x.nodes('/Shopping/ShoppingTrip/Item') as tbl(col)

select \* from tblTemp

drop table tblTemp

--for let where order by return

# 40. notes using table (shredding a table)

declare @x1 xml, @x2 xml

set @x1='<Shopping ShopperName="Phillip Burton" >

<ShoppingTrip ShoppingTripID="L1" >

<Item Cost="5">Bananas</Item>

<Item Cost="4">Apples</Item>

<Item Cost="3">Cherries</Item>

</ShoppingTrip></Shopping>'

set @x2='<Shopping ShopperName="Phillip Burton" >

<ShoppingTrip ShoppingTripID="L2" >

<Item>Emeralds</Item>

<Item>Diamonds</Item>

<Item>Furniture</Item>

</ShoppingTrip>

</Shopping>'

drop table #tblXML

create table #tblXML(pkXML INT PRIMARY KEY, xmlCol XML)

insert into #tblXML(pkXML, xmlCol) VALUES (1, @x1)

insert into #tblXML(pkXML, xmlCol) VALUES (2, @x2)

select \* from #tblXML

select tbl.col.value('@Cost','varchar(50)')

from #tblXML CROSS APPLY

xmlCol.nodes('/Shopping/ShoppingTrip/Item') as tbl(col)

# 41. Importing and exporting XML using the bcp utility

bcp [70-461S5].dbo.tblDepartment out mydata.out -N -T

create table dbo.tblDepartment2

([Department] varchar(19) null,

[DepartmentHead] varchar(19) null)

bcp [70-461S5].dbo.tblDepartment2 in mydata.out -N –T

# 42. Bulk Insert and Openrowset

drop table #tblXML

go

create table #tblXML(XmlCol xml)

go

bulk insert #tblXML from 'C:\XML\SampleDataBulkInsert.txt'

select \* from #tblXML

drop table #tblXML

go

create table #tblXML(IntCol int, XmlCol xml)

go

insert into #tblXML(XmlCol)

select \* from

openrowset(BULK 'C:\XML\SampleDataOpenRowset.txt', SINGLE\_BLOB) AS x

select \* from #tblXML

# 43. Schema

select E.EmployeeNumber, E.EmployeeFirstName, E.EmployeeLastName

, T.Amount, T.DateOfTransaction

from [dbo].[tblEmployee] as E

left join [dbo].[tblTransaction] as T

on E.EmployeeNumber = T.EmployeeNumber

where E.EmployeeNumber between 200 and 202

for xml raw, xmlschema --, xmldata

|  |  |  |
| --- | --- | --- |
| i4 or int | Int | Whole number, [integer](https://en.wikipedia.org/wiki/Integer) |
| Boolean |  | [Boolean](https://en.wikipedia.org/wiki/Boolean_datatype) logical value (0 or 1) |
| dateTime.iso8601 | Datetime | Date and time in [ISO 8601](https://en.wikipedia.org/wiki/ISO_8601)format |
| Double |  | [Double precision](https://en.wikipedia.org/wiki/Double_precision) floating point number |
| String | Varchar | String of characters. Must follow [XML encoding](https://en.wikipedia.org/wiki/XML#Characters_and_escaping). |
| Nil | Null | [Discriminated null value](https://en.wikipedia.org/wiki/Nullable_type); an XML-RPC [extension](https://web.archive.org/web/20050911054235/http:/ontosys.com/xml-rpc/extensions.php) |

# 46. XML Indexes

declare @x1 xml, @x2 xml

set @x1='<Shopping ShopperName="Phillip Burton" >

<ShoppingTrip ShoppingTripID="L1" >

<Item Cost="5">Bananas</Item>

<Item Cost="4">Apples</Item>

<Item Cost="3">Cherries</Item>

</ShoppingTrip></Shopping>'

set @x2='<Shopping ShopperName="Phillip Burton" >

<ShoppingTrip ShoppingTripID="L2" >

<Item>Emeralds</Item>

<Item>Diamonds</Item>

<Item>Furniture

<Color></Color></Item>

</ShoppingTrip>

</Shopping>'

drop table #tblXML;

create table #tblXML(pkXML INT PRIMARY KEY, xmlCol XML)

insert into #tblXML(pkXML, xmlCol) VALUES (1, @x1)

insert into #tblXML(pkXML, xmlCol) VALUES (2, @x2)

create primary xml index pk\_tblXML on #tblXML(xmlCol)

create xml index secpk\_tblXML\_Path on #tblXML(xmlCol)

using xml index pk\_tblXML FOR PATH

create xml index secpk\_tblXML\_Value on #tblXML(xmlCol)

using xml index pk\_tblXML FOR VALUE

create xml index secpk\_tblXML\_Property on #tblXML(xmlCol)

using xml index pk\_tblXML FOR PROPERTY