XSL Transformation (XSLT)

XSL Transformation

- XSLT stands for Extensible Stylesheet Transformation. It is an XML based language to transform source XML document to another form of documents i.e. XML, HTML etc.
- XSLT uses XPATH expression (will be discussed later) to traverse through the XML document.
- Steps to transform source XML to Target XML using XSL Transformation:

1. Declare XSL Namespace – XSL namespace should be declared at the beginning of the XSL stylesheet. A sample would like this:

   <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

2. Create XSL Stylesheet: It's about developing the actual transformation between the Source XML to Target XML. It's start with defining the output method, encoding style and indentation parameter followed by creating the transformation using different XSL Elements or functions.

XSL Elements	Description
<xsl:output>	To define the output method, encoding style and indentation parameter etc. e.g. **<xsl:output method="xml" encoding="utf-8" indent="no"/>**
<xsl:template>	This element is used to specify which part of the source XML should be transformed with the transformation rules being defined match attribute is used with XPATH expression to specify the part of the source XML to be considered for transformation. e.g. **<xsl:template match="/">**
<xsl:value-of>	This element is used to retrieve the value from a specified element or attribute of an element in the source XML. e.g. The value of ORDER_NUMBER element of Source XML is retrieved and assigned to OrderNum element of the Target XML **<OrderNum>** **<xsl:value-of select="/DATA_DS/G_1/ORDER_NUMBER"/>** **</OrderNum>**

XSL Transformation (XSLT)

XSL Elements	Description
<xsl:for-each>	This element is used iterate through a specified node. <xsl:value-of> gives value of a particular element but what if there are multiple instances of the same node, then to get all the values <xsl:for-each> is used. **<xsl:for-each select="DATA_DS/G_1">** **<SalesOrder>** **<OrderNum>** **<xsl:value-of select="ORDER_NUMBER" />** **</OrderNum>** **</xsl:for-each>**
<xsl:if>	Conditional IF statement in XSL. e.g. **<xsl:if test="OrderType='STANDARD'">** **</xsl:if>**
<xsl:choose>	This element is nothing but equivalent to the switch case(IF-ELSIF-ELSE) with one or many <xsl:when> and one <xsl:otherwise> to evaluate multiple conditions. e.g. **<xsl:choose>** **<xsl:when "InvoiceType='DEBITMEMO'">** **<xsl:value-of select='ATTRIBUTE1'/>** **</xsl:when>** **<xsl:otherwise>** **<xsl:value-of select='ATTRIBUTE2'/>** **</xsl:otherwise>** **</xsl:choose>**
<xsl:sort>	This element is used to sort the data based on a field value in the source xml. e.g. **<xsl:sort select='NAME' order='descending' data-type='text'/>**

3. Apply the XSLT on the source XML to transform source to the target XML – There are online platforms or tools available to test XSL code and transform source XML to Target XML. It is done programmatically in real time.

XML PATH (XPATH)

XPATH Syntaxes

XPATH Expression	Description
nodename	select all nodes with the “nodename”
/	select from root node.
//	select all the nodes being anywhere in the document from the current node that matches the node name provided
.	select current node
..	select parent of current node
@	select attributes

XPATH Predicates

Requirement	XPATH Expression
Select Name of the First Employee	**/employees/employee[1]/name[text()]**
Select the third employee element	**//employee[position()=3]**
Select the last employee element	**//employee[last()]**
Select all country element of the employees whose salary is greater than and equal to 25000	**//employee[salary>=25000]/country**
Select all employee elements belong to India	**//employee[country='India']**
Select employee whose id is 40	**//employee[@id='40']**

```
<employees>
  <employee id="10">
    <name>Leo Messi</name>
    <salary>10000</salary>
    <country>Argentina</country>
    <dept>Football</dept>
  </employee>
  <employee id="20">
    <name>Ronaldo</name>
    <salary>20000</salary>
    <country>Portugal</country>
    <dept>Football</dept>
  </employee>
  <employee id="30">
    <name>Sachin Tendulkar</name>
    <salary>30000</salary>
    <country>India</country>
    <dept>Cricket</dept>
  </employee>
  <employee id="40">
    <name>Neeraj Kumar</name>
    <salary>25000</salary>
    <country>India</country>
    <dept>Javeline</dept>
  </employee>
</employees>
```