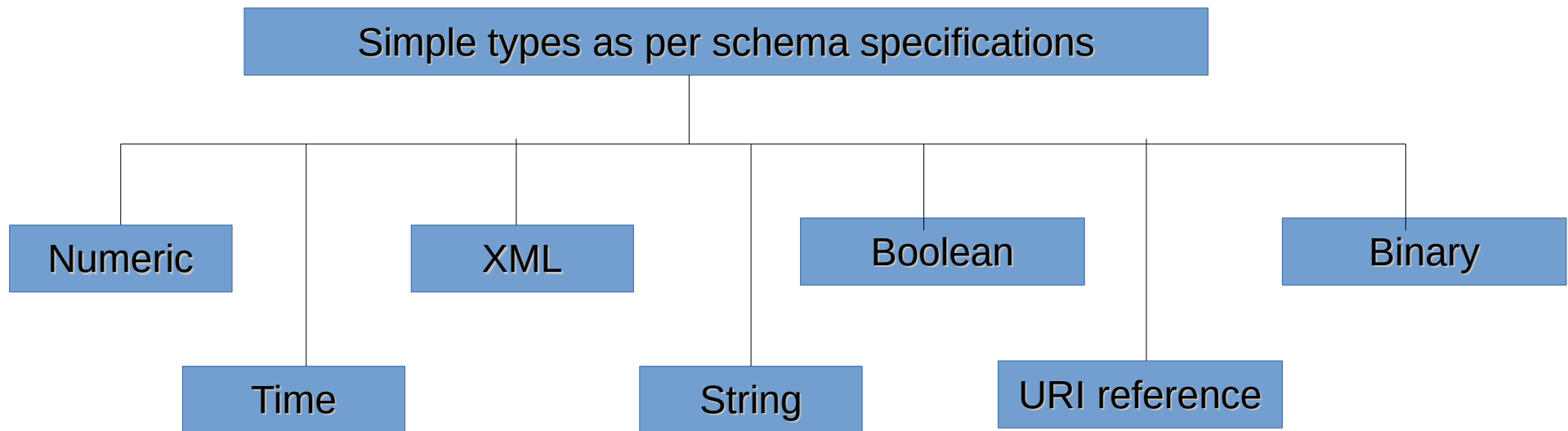


UNIT – 3 XML Schema

- Introduction to Schema
- Features
- DTD versus XML Schema
- XML Schema Type System
- Simple types
- Complex Types
- Grouping of Data
- Deriving Types
- Attributes

Simple Types



- XML schemas **offer 44 built-in simple types.**
- The XML schema simple type is **classify into seven categories.**

Simple Type – Numeric Data Types

Data Type Name	Meaning	Example
xsd:float	32 – bit float type	0, 12345.2356
xsd:double	64 – bit double type	0,45.89E-2
xsd:decimal	Arbitrary precision (Big Decimal)	87200.29, -3.124578926
xsd:integer	Arbitrary large or small number	-789253598138965, 24535989326475
xsd:nonPositiveInteger	Integer less than or equal to 0	0, -1, -2

Simple Type – Numeric Data Types

Data Type Name	Meaning	Example
xsd:negativeInteger	Integer less than 0	-1,-2,-3
xsd:nonNegativeInteger	Integer greater than or equal to 0	0,1, 2 ,3
xsd:positiveInteger	Integer greater than 0	1,2,3
xsd:long	8 byte 2's complement integer	2356985668522
xsd:int	4 byte 2's complement integer	-615251, 0, 125369

Simple Type – Numeric Data Types

Data Type Name	Meaning	Example
xsd:short	2 – byte , 2's complement integer	-32767 to +32767
xsd:byte	1 – byte 2's complement integer	-128 to +127
xsd:unsignedLong	8 byte unsigned long	
xsd:unsignedInt	4 byte unsigned integer	
xsd:unsignedShort	2 byte unsigned integer	
xsd:unsignedByte	1 byte unsigned integer	

Simple Type – Time Data Types

Data Type Name	Meaning	Example
xsd:dateTime	Date and time in the formate YYYY-MM-DDTHH:MM:SS	2006-02-25T06:05:33
xsd:date	YYYY-MM-DD	
xsd:time	HH:MM:SS	
xsd:gDay	A day in a month	--01, --23
xsd:gMonth	A month in a year	--02--, --05--
xsd:gYear	A Year	2006

Simple Type – Time Data Types

Data Type Name	Meaning	Example
Xsd:YearMonth	A pecific month in a specific year	2006-02, 2017-05
xsd:gMonthDay	A date without year	--01-20
xsd:duration	Length of time in format	P2006Y01M20DT06H11M03S

Simple Type – XML Data Types

Data Type Name	Meaning	Example
xsd:ID	A unique value for an element or an attribute	T1, M90, G101
xsd:IDREF	Value of another ID type defined elsewhere in the document	
xsd:ENTITY	An XML name, declared as an unpaired entity in DTD	
xsd:NOTATION	Usually indicates a file format	
xsd:IDREFS	Reference to a list of ID names	
xsd:ENTITIES	List of entity names	

Simple Type – XML Data Types

Data Type Name	Meaning	Example
xsd:NMTOKEN	NMTOKEN type	
xsd:NMTOKENS	A list of NMTOKEN types	
xsd:language	Language name from a list of valid values	
xsd:Name	XML name with or without colons	Student, emp
xsd:QName	Prefixed name	
xsd:NCName	Local name without colons	

Simple Type – String Data Types

Data Type Name	Meaning	Example
xsd:string	A unicode character based string of any length.	
xsd:normalized String	A string in which all the carriage returns, linefeeds, and tabs are replaced with a single blank (space) character.	
xsd:token	Same as above, but in addition all leading and trailing spaces are trimmed and consecutive spaces are converted into a single space.	

Simple Type – Binary Data Types

- XML support binary data types.
- But the problem with binary data is that it can have byte patterns that are illegal.
- **This is because some characters such as null have a different meaning, and they cannot be a part of the XML content.**
- We need to encode such illegal characters into a legal form
 - By hexadecimal conversion
 - By base-64 encoding

Simple Type – Other Data Types

- **Boolean Data Type**

- xsd:boolean

- it allows one of the four possible value : zero, one, true and false.

- **URI Data Type**

- xsd:URI

- It allows us to specify a URI
 - For i.e – <http://www.test.com/name.html>

Deriving Types

- There are three technique for deriving types

Deriving simple types in XML schemas

```
graph TD; A[Deriving simple types in XML schemas] --> B[Restriction]; A --> C[Union]; A --> D[List]
```

Restriction

Union

List

Deriving Types - Restriction

- **Restriction**

- Restriction allows us to select a subset of values allowed by the base type.
- We can use an element of type `xsd:restriction` as a child element of an `xsd:simple type` element.

Deriving Types - Restriction - Facets

- **Restriction - Facets**

- A facet allows us to specify more restrictions than what a basic type allows.

Facet	Description
xsd:minInclusive	The minimum value that all the instancees of this type must be grater than or equal to
xsd:maxInclusive	The maximum value that all the instancees of this type must be less than or equal to
xsd:minExclusive	The minimum value that all the instancees of this type must be grater than
xsd:maxExclusive	The maximum value that all the instancees of this type must be less than
xsd:enumeration	A list of allowed values
xsd:whiteSpace	How white spce is treatedin this element

Deriving Types - Restriction - Facets

- **Restriction - Facets**

- A facet allows us to specify more restrictions than what a basic type allows.

Facet	Description
xsd:pattern	A pattern with which the contents of the element are compared
xsd:length	The length of a string, items in a lis, or bytes in binary data
xsd:minLength	The minimum length
xsd:maxLength	The maximum length
xsd:totalDigits	The maximum number of digits allowed in the element
xsd:fractionDigits	The maximum number of digits allowed in the fractional part of the element.

Deriving Types - Restriction – Enumeration

- **Restriction - Enumeration**

- The enumeration facet in XML schemas allows us to specify a list of possible values for an element.

```
<xsd:simpleType name="BookCategory">  
  <xsd:restriction base="xsd:string">  
    <xsd:enumeration value="computer archi"/>  
    <xsd:enumeration value="network"/>  
  </xsd:restriction>  
</xsd:simpleType>
```

Attributes

- `<attribute name="des" type="xsd:type" use=""/>`
- An attribute occurs only once.
- We can specify whether
 - **Required** : must have
 - **Optional** : may have an attribute
 - **Prohibited** : cannot have an attribute
- i.e

`<attribute name="designation" type="xsd:string" use="required" default="Manager"/>`

Grouping Attributes

- If an element have several attributes, then we can group them and provide a reference of this group to the concerned element.

Grouping Attributes

```
<xsd:element name="EMPLOYEE">
```

```
  <xsd:complexType>
```

```
    <xsd:attributeGroup ref="empDetails"/>
```

```
  </xsd:complexType>
```

```
</xsd:element>
```

```
<xsd:attributeGroup name="empDetails">
```

```
  <xsd:attribute name="empID" use="required" type="xsd:ID"/>
```

```
  <xsd:attribute name="name" use="required" type="xsd:string"/>
```

```
  <xsd:attribute name="designation" use="optional" type="xsd:string">
```

```
</xsd:attributeGroup>
```

Features of Schema

- XML schema uses XML instance syntax
- XML schema allows a rich variety of data types to be used to constrain both element and attribute content.
- XML schema allows us to specify which namespace declarations and definitions belongin.
- In XML schema, type definitions and element and attribute declarations are separated from each other.
- XML schema allows us to specify constraints on the uniqueness of values of a particular type, as well as relationships between those unique values and value of other type.