XML Schema(2) Reference

Defining uniqueness (ID type)

```
<students xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="slide2.xsd">
  <student number="x1234567">James Bond</student>
  <student number="x1234568">Harry Potter</student>
</students>
```

"slide2.xsd"

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

   <xsd:element name="students" type="studentsType"/>

   <xsd:complexType name="studentsType">
     <xsd:sequence>
       <xsd:element name="student" type="studentType"
                                   minOccurs="1" maxOccurs="unbounded"/>
     </xsd:sequence>
   </xsd:complexType>

   <xsd:complexType name="studentType">
      <xsd:simpleContent>
         <xsd:extension base="xsd:string">
            <xsd:attribute name="number" type="xsd:ID"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>

</xsd:schema>
```

Definition of element students

Definition of element student

Many occurrencies of element student

Identifier type

Defining uniqueness (unique)

```
<students xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="slide3.xsd">
  <student>
    <number>1234567</number>
    <name>James Bond</name>
  </student>
   <student>
    <number>1234568</number>
    <name>Harry Potter</name>
  </student>
 <student>
    <name>Spiderman</name>
  </student>
</students>
```

"slide4.xsd"
(... next slide ...)

Defining uniqueness (unique)

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

   <xsd:element name="students">
     <xsd:complexType>
       <xsd:sequence>
         <xsd:element name="student" type="studentType"
                                     minOccurs="1" maxOccurs="unbounded"/>
       </xsd:sequence>
     </xsd:complexType>
     <xsd:unique name="student">
       <xsd:selector xpath="student"/>
       <xsd:field xpath="number"/>
     </xsd:unique>
   </xsd:element>

   <xsd:complexType name="studentType">
      <xsd:sequence>
         <xsd:element name="number" type="xsd:nonNegativeInteger"/
                                    minOccurs="0"/>
         <xsd:element name="name" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>

</xsd:schema>
```

Definition of element students

Definition of element student

Name of unique constraint

Path to unique element number

Value of unique element number

Element number can be empty

Defining uniqueness (single key)

```
<students xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="slide3.xsd">
  <student>
    <number>1234567</number>
    <name>James Bond</name>
  </student>
   <student>
    <number>1234568</number>
    <name>Harry Potter</name>
  </student>
 <student>
    <number>1234569</number>
    <name>Spiderman</name>
  </student>
</students>
```

"slide6.xsd"
(... next slide ...)

Defining uniqueness (single key)

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

   <xsd:element name="students">
     <xsd:complexType>
       <xsd:sequence>
         <xsd:element name="student" type="studentType"
                                     minOccurs="1" maxOccurs="unbounded"/>
       </xsd:sequence>
     </xsd:complexType>
     <xsd:key name="student">
       <xsd:selector xpath="student"/>
       <xsd:field xpath="number"/>
     </xsd:key>
   </xsd:element>

   <xsd:complexType name="studentType">
      <xsd:sequence>
         <xsd:element name="number" type="xsd:nonNegativeInteger"
                                    minOccurs="1"/>
         <xsd:element name="name" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>

</xsd:schema>
```

Definition of element students

Definition of element student

Name of key constraint

Path to key element number

Value of unique element number

Element number cannot be empty

Defining uniqueness (composite key)

```
<rooms xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="slide8.xsd">
  <room>
    <bnumber>3</bnumber>
    <rnumber>210</rnumber>
    <area>23.5</area>
  </room>
 <room>
    <bnumber>4</bnumber>
    <rnumber>210</rnumber>
    <area>34.6</area>
  </room>
</rooms>
```

"slide8.xsd"
(... next slide ...)

Defining uniqueness (composite key)

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSc
    <xsd:element name="rooms">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="room" type="roomType"
                                   minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:key name="room">
        <xsd:selector xpath="room"/>
        <xsd:field xpath="bnumber"/>
        <xsd:field xpath="rnumber"/>
      </xsd:key>
    </xsd:element>

    <xsd:complexType name="roomType">
       <xsd:sequence>
          <xsd:element name="bnumber" type="xsd:nonNegativeInteger"/>
          <xsd:element name="rnumber" type="xsd:nonNegativeInteger"/>
          <xsd:element name="area" type="xsd:number"/>
       </xsd:sequence>
    </xsd:complexType>

</xsd:schema>
```

Definition of element rooms
Definition of element room
Name of key constraint
Path to key elements (bnumber, rnumber)
Value of unique elements (bnumber, rnumber)

Defining references (referencing element)

```
<enrolments xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="slide10-11.xsd">
  <student>
    <number>1234567</number>
    <name>James Bond</name>
  </student>
   <student>
    <number>1234568</number>
    <name>Harry Potter</name>
  </student>
  <subject>
    <code>CSCI235</code>
    <enroled-by>1234568</enroled-by>
  </subject>
  <subject>
    <code>ITCS206</code>
    <enroled-by>1234567</enroled-by>
  </subject>
</enrolments>
```

"slide10-11.xsd"
(... next slide ...)

Defining references (referencing element)

```
<xsd:schema xmlns:xsd="http://www.w3.org/
    <xsd:element name="enrolments">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="student" type="studentType"
                                   minOccurs="1" maxOccurs="unbounded"/>
          <xsd:element name="subject" type="subjectType"
                                   minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
      <xsd:key name="student">
        <xsd:selector xpath="student"/>
        <xsd:field xpath="number"/>
      </xsd:key>
      <xsd:keyref name="student-ref" refer="student">
        <xsd:selector xpath="subject"/>
        <xsd:field xpath="enroled-by"/>
      </xsd:keyref>
    </xsd:element>
...
```

Definition of element enrolments
Definition of element student
Definition of element subject
Name of key constraint
Path to key element number
Value of key element number
Name of referenced key constraint
Path to key element subject
Value of element subject

Defining references (referencing element)

```
...
 <xsd:complexType name="studentType">
       <xsd:sequence>
          <xsd:element name="number" type="xsd:nonNegativeInteger"
                                     minOccurs="1"/>
          <xsd:element name="name" type="xsd:string"/>
       </xsd:sequence>
    </xsd:complexType>

 <xsd:complexType name="subjectType">
       <xsd:sequence>
          <xsd:element name="code" type="xsd:string"/>
          <xsd:element name="enroled-by" type="xsd:nonNegativeInteger"
                                     minOccurs="1" maxOccurs="unbounded"/>
       </xsd:sequence>
    </xsd:complexType>

</xsd:schema>
```

Definition of studentType
Definition of subjectType

Defining references (referencing attribute)

```
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide13-14.xsd" >
  <students>
    <student number="x1234567"/>
    <student number="x1234568"/>
  </students>
  <enrolments>
    <enrolment subject="CSCI235" enroled-by="x1234567"/>
    <enrolment subject="CSCI235" enroled-by="x1234568"/>
    <enrolment subject="ITCS206" enroled-by="x1234568"/>
  </enrolments>
</root>
```

"slide13-14.xsd"
(... next slide ...)

Defining references (referencing attribute)

```
<xsd:schema xmlns:xsd="http://www.w3.org/1
  <xsd:element name="root">
    <xsd:complexType>
      <xsd:sequence minOccurs="1" ma
        <xsd:element name="students" type="studentsType"/>
        <xsd:element name="enrolments" type="enrolmentsType"/>
      </xsd:sequence>
    </xsd:complexType>
    <xsd:key name="studentId">
      <xsd:selector xpath="./students/stude
      <xsd:field xpath="@number"/>
    </xsd:key>
    <xsd:keyref name="referenceId" refer="studentId">
      <xsd:selector xpath="./enrolments/enrolment"
      <xsd:field xpath="@enroled-by"/>
    </xsd:keyref>
  </xsd:element>
...
```

Definition of element root

Definition of element student

Definition of element enrolments

Name of key constraint

Path to element student

Value of key attribute number

Name of referenced key constraint

Value of element enrolment

Path to element enrolment

Defining references (referencing attribute)

```
...
 <xsd:complexType name="studentsType">
    <xsd:sequence minOccurs="1">
      <xsd:element name="student" minOccurs="1" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:attribute name="number" type="xsd:NCName" use="required"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
 </xsd:complexType>

 <xsd:complexType name="enrolmentsType">
    <xsd:sequence minOccurs="1">
      <xsd:element name="enrolment" minOccurs="1" maxOccurs="unbounded">
        <xsd:complexType>
         <xsd:attribute name="subject" type="xsd:string" use="required"/>
          <xsd:attribute name="enroled-by" type="xsd:NCName" use="required"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
 </xsd:complexType>
</xsd:schema>
```

Key attribute number

Referencing attribute enroled-by

Defining references (IDREF attribute)

```
<root xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="16-17.xsd" >
  <students>
    <student number="x1234567"/>
    <student number="x1234568"/>
  </students>
  <enrolments>
    <enrolment subject="CSCI235" enroled-by="x1234567"/>
    <enrolment subject="CSCI235" enroled-by="x1234568"/>
    <enrolment subject="ITCS206" enroled-by="x1234568"/>
  </enrolments>
</root>
```

"slide16-17.xsd"
(... next slide ...)

Defining references (IDREF attribute)

```
<xsd:schema xmlns:xsd="http://www.w3.org/2
  <xsd:element name="root">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="students" type="studentsType"/>
        <xsd:element name="enrolments" type="enrolmentsType"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
...
```

Definition of element root

Definition of element student

Definition of element enrolments

Defining references (IDREF attribute)

```
...
 <xsd:complexType name="studentsType">
    <xsd:sequence>
      <xsd:element name="student" minOccurs="1" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:attribute name="number" type="xsd:ID"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
 </xsd:complexType>

 <xsd:complexType name="enrolmentsType">
    <xsd:sequence>
      <xsd:element name="enrolment" minOccurs="1" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:attribute name="subject" type="xsd:string"/>
          <xsd:attribute name="enroled-by" type="xsd:IDREF"/>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
 </xsd:complexType>
</xsd:schema>
```

Definition of attribute number of type ID

Definition of attribute enroled-by of type IDREF

Using namespaces

```
<industry:company
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
  xmlns:industry="http://www.industry.com"
  xsi:schemaLocation="http://www.industry.com slide18.xsd">
  Golden Bolts Pty. Ltd.
</industry:company>
```

"slide18.xsd"

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLS
            targetNamespace="http://www.industry.com"
            elementFormDefault="qualified"
            attributeFormDefault="unqualified"
            xmlns="http://www.industry.com">

   <xsd:element name="company" type="companyType"/>

   <xsd:simpleType name="companyType">
     <xsd:restriction base="xsd:string">
       <xsd:minLength value="1"/>
       <xsd:maxLength value="60"/>
     </xsd:restriction>
   </xsd:simpleType>

</xsd:schema
```

industry namespace

Location of schema

Target namespace= document namespace

Elements will be qualified with default namespace prefix

Default namespace

Using namespaces

```
<industry:company
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
  xmlns:industry="http://www.industry.com"
  xsi:schemaLocation="http://www.industry.com slide19.xsd">
  Golden Bolts Pty. Ltd.
</industry:company>
```

industry namespace

Location of schema

"slide19.xsd"

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSch
            targetNamespace="http://www.industry.com"
            elementFormDefault="qualified"
            attributeFormDefault="unqualified"
            xmlns:ind="http://www.industry.com">

    <xsd:element name="company" type="ind:companyType"/>

    <xsd:simpleType name="companyType">
      <xsd:restriction base="xsd:string">
        <xsd:minLength value="1"/>
        <xsd:maxLength value="30"/>
      </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
```

Target namespace= document namespace

Elements will be qualified with ind namespace prefix

Attributes will not be qualified with ind namespace prefix

ind namespace

Qualification of companyType

Using namespaces

```
<my:camera xmlns:my="http://www.camera.org"
           xmlns:nikon="http://www.nikon.com"
           xmlns:pentax="http://www.pentax.com"
           xmlns:olympus="http://www.olympus.com"
           xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
           xsi:schemaLocation="http://www.camera.org slide21.xsd">
        <body>
            <nikon:description>
                  Ergonomically designed casing for easy handling
            </nikon:description>
        </body>
        <lens>
            <olympus:zoom>300mm</olympus:zoom>
            <olympus:f-stop>1.2</olympus:f-stop>
        </lens>
        <manual-adapter>
            <pentax:speed>1/10,000 sec to 100 sec</pentax:speed>
            <pentax:description>Very high shutter speed</pentax:description>
        </manual-adapter>
 </my:camera>
```

"slide21.xsd"
(... next slide ...)

Using namespaces

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchem
            targetNamespace="http://www.camera.org"
            xmlns="http://www.camera.org"
            xmlns:nikon="http://www.nikon.com"
            xmlns:olympus="http://www.olympus.com"
            xmlns:pentax="http://www.pentax.com"
            elementFormDefault="unqualified">
    <xsd:import namespace="http://www.nikon.com" schemaLocation="nikon.xsd"/>
    <xsd:import namespace="http://www.olympus.com" schemaLocation="olympus.xsd"/>
    <xsd:import namespace="http://www.pentax.com" schemaLocation="pentax.xsd"/>
    <xsd:element name="camera">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="body" type="nikon:body"/>
                <xsd:element name="lens" type="olympus:lens"/>
                <xsd:element name="manual-adapter" type="pentax:manual-adapter"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

nikon namespace

olympus namespace

pentax namespace

Qualification of body

Qualification of lens

Qualification of pentax

"nikon.xsd","olympus.xsd","pentax.xsd"
(... next slides ...)

Using namespaces

"nikon.xsd"

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://www.nikon.com"
            xmlns="http://www.nikon.com"
            elementFormDefault="qualified">

        <xsd:complexType name="body">
            <xsd:sequence>
                <xsd:element name="description" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
</xsd:schema>
```

Definition of body type in nikon namespace

Using namespaces

"olympus.xsd"

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://www.olympus.com"
            xmlns="http://www.olympus.com"
            elementFormDefault="qualified">

        <xsd:complexType name="lens">
            <xsd:sequence>
                <xsd:element name="zoom" type="xsd:string"/>
                <xsd:element name="f-stop" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
</xsd:schema>
```

Definition of lens type in olympus namespace

Using namespaces

"pentax.xsd"

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSche
            targetNamespace="http://www.pentax.com"
            xmlns="http://www.pentax.com"
            elementFormDefault="qualified">

        <xsd:complexType name="manual-adapter">
            <xsd:sequence>
                <xsd:element name="speed" type="xsd:string"/>
                <xsd:element name="description" type="xsd:string"/>
            </xsd:sequence>
        </xsd:complexType>
</xsd:schema>
```

Definition of manual-adatpter type in pentax namespace

Using global elements

```
<students xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:noNamespaceSchemaLocation="slide25.xsd">
  <student>James Bond</student>
  <student>Harry Potter</student>
</students>
```

Global element `student`

"slide25.xsd"

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <xsd:element name="student" type="xsd:string"/>

    <xsd:element name="students" type="studentsType"/>

    <xsd:complexType name="studentsType">
      <xsd:sequence>
        <xsd:element ref="student" minOccurs="1"
                                   maxOccurs="unbounded"/>
      </xsd:sequence>
    </xsd:complexType>

</xsd:schema>
```

Reference to a global element `student`

Using global elements

```
<student xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="slide26.xsd">
         number="x1234567">James Bond</student>
```

Global attribute `number`

"slide26.xsd"

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <xsd:attribute name="number" type="xsd:ID"/>

    <xsd:element name="student" type="studentType"/>

    <xsd:complexType name="studentType">
       <xsd:simpleContent>
          <xsd:extension base="xsd:string">
             <xsd:attribute ref="number"/>
          </xsd:extension>
       </xsd:simpleContent>
    </xsd:complexType>

</xsd:schema>
```

Reference to a global attribute `number`

Using global elements: method 1

```
<library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="slide28.xsd">
  <book>
    <title>Databases</title>
    <author>Harry Potter</author>
  </book>
  <book>
    <title>XML</title>
    <author>James Bond</author>
  </book>
</library>
```

"slide28.xsd"
(... next slide ...)

Using global elements: method 1

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <xsd:element name="title" type="xsd:string"/>
    <xsd:element name="author" type="xsd:string"/>

    <xsd:element name="library">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="book" maxOccurs="unbounded">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element ref="title"/>
                <xsd:element ref="author"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>

</xsd:schema>
```

Global element `title`

Global element `author`

Reference to a global element `title`

Reference to a global element `author`

Using global elements: method 2

```
<library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="slide30.xsd">
  <book>
    <title>Databases</title>
    <author>Harry Potter</author>
  </book>
  <book>
    <title>XML</title>
    <author>James Bond</author>
  </book>
</library>
```

"slide30.xsd"
(... next slide ...)

Using global elements: method 2

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <xsd:element name="title" type="xsd:string"/>
    <xsd:element name="author" type="xsd:string"/>

    <xsd:element name="book">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="title"/>
          <xsd:element ref="author"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>

    <xsd:element name="library">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element ref="book" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Global element `title`

Global element `author`

Global element `book`

References to `title` and `author`

Reference to `book`

Using global elements: method 3

```
<library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="slide32.xsd">
  <book>
    <title>Databases</title>
    <author>Harry Potter</author>
  </book>
  <book>
    <title>XML</title>
    <author>James Bond</author>
  </book>
</library>
```

"slide32.xsd"
(... next slide ...)

Using global elements: method 3

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <xsd:element name="title" type="xsd:string"/>
    <xsd:element name="author" type="xsd:string"/>

      <xsd:complexType name="bookType">
        <xsd:sequence>
          <xsd:element ref="title"/>
          <xsd:element ref="author"/>
        </xsd:sequence>
      </xsd:complexType>

    <xsd:element name="library">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="book" type="bookType" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Global element title

Global element author

Definition of type bookType

References to title and author

Definition of element book

Using global elements: method 4

```
<library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="slide32.xsd">
  <book>
    <title>Databases</title>
    <author>Harry Potter</author>
  </book>
  <book>
    <title>XML</title>
    <author>James Bond</author>
  </book>
</library>
```

"slide34.xsd"
(... next slide ...)

Using global elements: method 4

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <xsd:element name="title" type="xsd:string"/>
    <xsd:element name="author" type="xsd:string"/>

      <xsd:group name="bookGroup">
        <xsd:sequence>
          <xsd:element name="book">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element ref="title"/>
                <xsd:element ref="author"/>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:group>

    <xsd:element name="library">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:group ref="bookGroup" maxOccurs="unbounded"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

Global element title

Global element author

Global group bookGroup

References to title and author

Reference to bookGroup

One more example

```
<company xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://www.company.org slide36.xsd"
         xmlns="http://www.company.org">

 <person>
   <name>James Bond</name>
   <ssn>007</ssn>
  </person>
  <person>
    <name>Harry Potter</name>
    <ssn>666</ssn>
  </person>

  <product>
    <name>bolt</name>
  </product>
  <product>
    <name>screw</name>
  </product>
</company>
```

"slide36.xsd"
(... next slide ...)

One more example ...

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            targetNamespace="http://www.company.org"
            xmlns="http://www.company.org"
            elementFormDefault="qualified">
    <xsd:include schemaLocation="person.xsd"/>
    <xsd:include schemaLocation="product.xsd"/>
    <xsd:element name="company">
        <xsd:complexType>
            <xsd:sequence>
                <xsd:element name="person" type="personType"
                                           maxOccurs="unbounded"/>
                <xsd:element name="product" type="productType"
                                           maxOccurs="unbounded"/>
            </xsd:sequence>
        </xsd:complexType>
    </xsd:element>
</xsd:schema>
```

A file where type declaration is

A file where type declaration is

Derived type

Derived type

"person.xsd, "product.xsd"
(... next slide ...)

One more example …

"person.xsd"

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            elementFormDefault="qualified">
    <xsd:complexType name="personType">
        <xsd:sequence>
           <xsd:element name="name" type="xsd:string"/>
           <xsd:element name="ssn" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```

"product.xsd"

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            elementFormDefault="qualified">
    <xsd:complexType name="productType">
        <xsd:sequence>
           <xsd:element name="name" type="xsd:string"/>
        </xsd:sequence>
    </xsd:complexType>
</xsd:schema>
```