XML Schema(1) Reference

Schemas and schema languages

A schema is a definition of the structures of XML documents

A schema language is a formal language for expressing schemas

XML document is valid if its structures are consistent with the structures defined in its schema

Schema processing: given an XML document and a schema, a schema processor checks for validity, i.e. that the document conforms to the schema requirements

If the document is valid then its normalized version is created: default attributes and elements are inserted

Schemas are similar to grammars for programming languages

Proposals for schema languages

W3C proposals:

- DTD
- XML-Data, January 1998
- DCD (Document Content Description), July 1998
- DDML (Document Definition Markup Language), Jan 1999
- SOX (Schema for Object-oriented XML), July 1999
- XML Schema

Non-W3C proposals:

- Assertion Grammars by Dave Raggett
- Schematron by Rick Jellife
- TREX (Tree Regular Expressions for XML) by James Clark
- Examplotron by Eric van der Vlist
- RELAX by Makoto Murara / RELAX NG by Murata and Clark
- DSD (Document Structure Description)

XML Schema ? What is it ?

XML Schema is a formal notation for defining a schema for a class of XML documents

XML Schema is a vocabulary for expressing data dependencies

XML Schema document models are designed to define the usage and relationships of various schema components, such as the following:

- Datatypes
- Elements and their content
- Attributes and their values
- Reusable components and their content
- Notations

Example

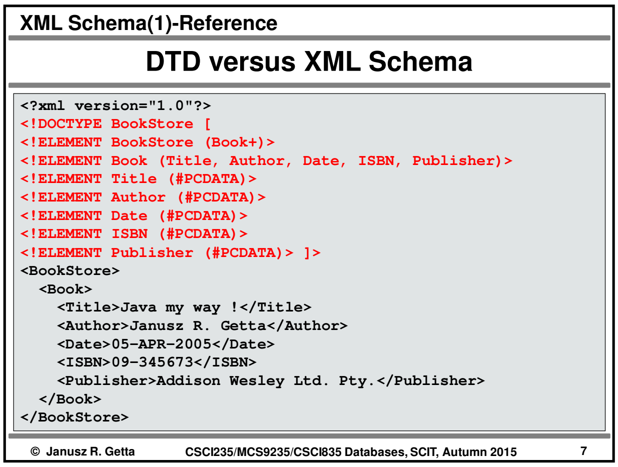
Is the following data valid ?

```
<location>
          <latitude>32.904237</latitude>
          <longitude>73.620290</longitude>
          <uncertainty units="meters">2</uncertainty>
</location>
```

XML Schema: an overview

Properties:

- Enhanced data types (44+)
- Type definition mechanisms
- The same syntax as XML documents
- Object-orientation (inheritance)
- Create type constructor
- Ability to specify uniqueness (keys or content) constraints
- Ability to define the multiple elements with the same name but different content
- Ability to define the elements with nil contents
- Ability to define substitutable elements, e.g. the "Book" element is substitutable for the "Publication" element.

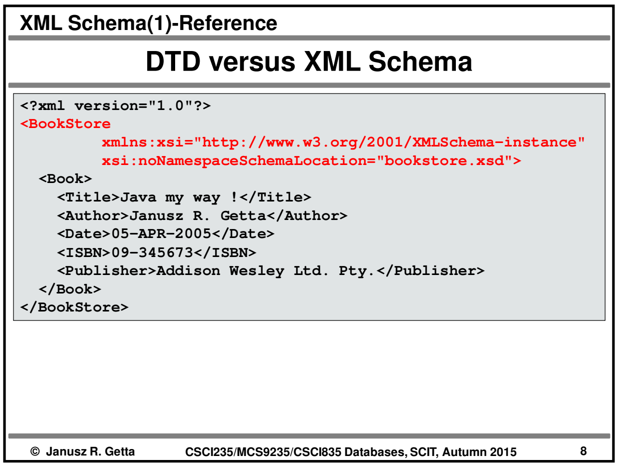


XML Schema(1)-Reference

DTD versus XML Schema

```
<?xml version="1.0"?>
<!DOCTYPE BookStore [
<!ELEMENT BookStore (Book+)>
<!ELEMENT Book (Title, Author, Date, ISBN, Publisher)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT Author (#PCDATA)>
<!ELEMENT Date (#PCDATA)>
<!ELEMENT ISBN (#PCDATA)>
<!ELEMENT Publisher (#PCDATA)> ]>
<BookStore>
  <Book>
    <Title>Java my way !</Title>
    <Author>Janusz R. Getta</Author>
    <Date>05-APR-2005</Date>
    <ISBN>09-345673</ISBN>
    <Publisher>Addison Wesley Ltd. Pty.</Publisher>
  </Book>
</BookStore>
```

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 7

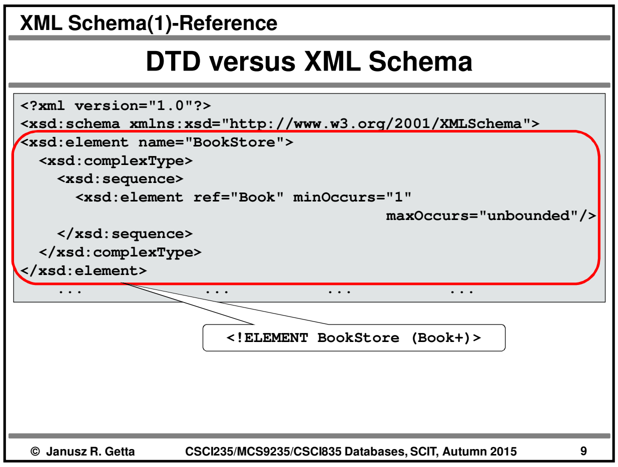


XML Schema(1)-Reference

DTD versus XML Schema

```
<?xml version="1.0"?>
<BookStore
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="bookstore.xsd">
  <Book>
    <Title>Java my way !</Title>
    <Author>Janusz R. Getta</Author>
    <Date>05-APR-2005</Date>
    <ISBN>09-345673</ISBN>
    <Publisher>Addison Wesley Ltd. Pty.</Publisher>
  </Book>
</BookStore>
```

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 8



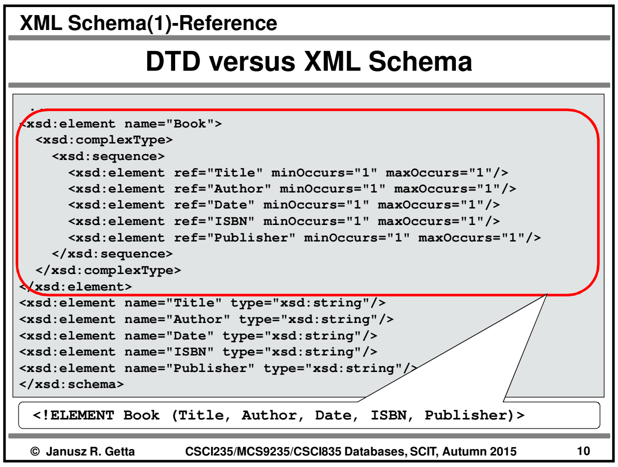
XML Schema(1)-Reference

DTD versus XML Schema

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<xsd:element name="BookStore">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Book" minOccurs="1"
                                  maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
    ...             ...          ...          ...
```

```
<!ELEMENT BookStore (Book+)>
```

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 9



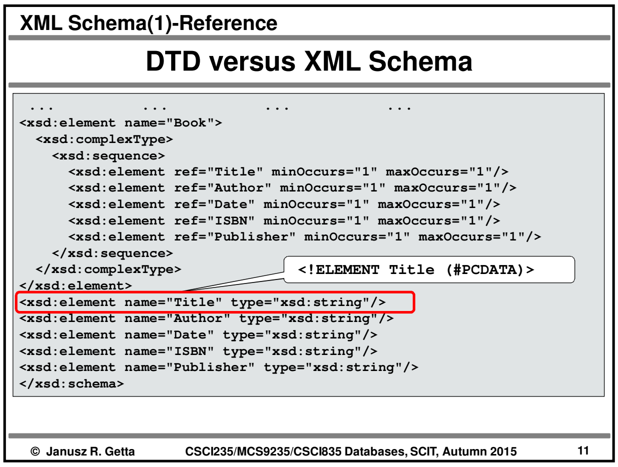
XML Schema(1)-Reference

DTD versus XML Schema

```
 ...
<xsd:element name="Book">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Title" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Author" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Date" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="ISBN" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Publisher" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Title" type="xsd:string"/>
<xsd:element name="Author" type="xsd:string"/>
<xsd:element name="Date" type="xsd:string"/>
<xsd:element name="ISBN" type="xsd:string"/>
<xsd:element name="Publisher" type="xsd:string"/>
</xsd:schema>
```

```
<!ELEMENT Book (Title, Author, Date, ISBN, Publisher)>
```

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 10



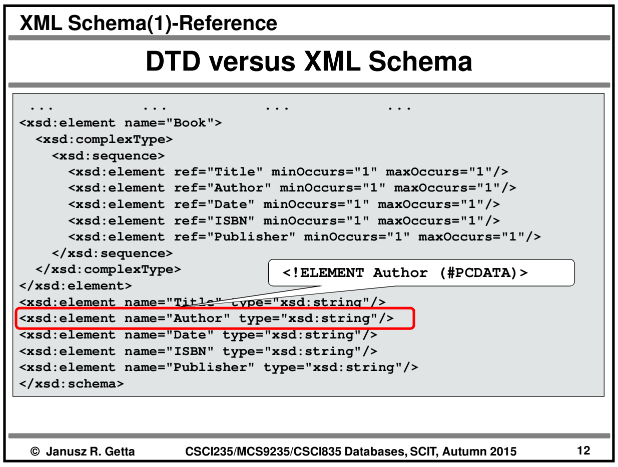
XML Schema(1)-Reference

DTD versus XML Schema

```
 ...           ...            ...            ...
<xsd:element name="Book">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Title" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Author" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Date" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="ISBN" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Publisher" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Title" type="xsd:string"/>
<xsd:element name="Author" type="xsd:string"/>
<xsd:element name="Date" type="xsd:string"/>
<xsd:element name="ISBN" type="xsd:string"/>
<xsd:element name="Publisher" type="xsd:string"/>
</xsd:schema>
```

```
<!ELEMENT Title (#PCDATA)>
```

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 11



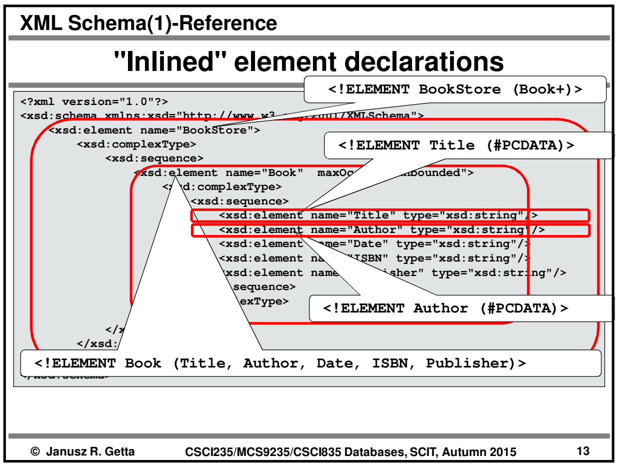
XML Schema(1)-Reference

DTD versus XML Schema

```
 ...           ...            ...            ...
<xsd:element name="Book">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="Title" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Author" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Date" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="ISBN" minOccurs="1" maxOccurs="1"/>
      <xsd:element ref="Publisher" minOccurs="1" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="Title" type="xsd:string"/>
<xsd:element name="Author" type="xsd:string"/>
<xsd:element name="Date" type="xsd:string"/>
<xsd:element name="ISBN" type="xsd:string"/>
<xsd:element name="Publisher" type="xsd:string"/>
</xsd:schema>
```

```
<!ELEMENT Author (#PCDATA)>
```

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 12



XML Schema(1)-Reference

"Inlined" element declarations

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.   /2001/XMLSchema">
   <xsd:element name="BookStore">
       <xsd:complexType>
           <xsd:sequence>
               <xsd:element name="Book"  maxOc      nbounded">
                   < d:complexType>
                       <xsd:sequence>
                           <xsd:element name="Title" type="xsd:string"/>
                           <xsd:element name="Author" type="xsd:string"/>
                           <xsd:element  me="Date" type="xsd:string"/>
                           <xsd:element na  "ISBN" type="xsd:string"/>
                           xsd:element name    isher" type="xsd:string"/>
                            sequence>
                            exType>

           </x
       </xsd:
```

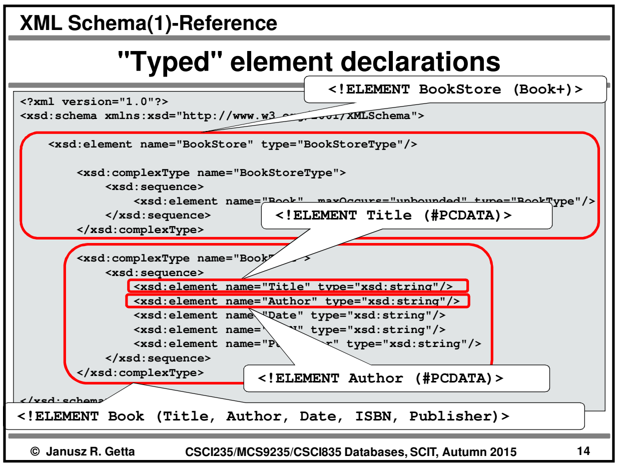
<!ELEMENT BookStore (Book+)>

<!ELEMENT Title (#PCDATA)>

<!ELEMENT Author (#PCDATA)>

<!ELEMENT Book (Title, Author, Date, ISBN, Publisher)>

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 13



XML Schema(1)-Reference

"Typed" element declarations

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3     /XMLSchema">

   <xsd:element name="BookStore" type="BookStoreType"/>

       <xsd:complexType name="BookStoreType">
           <xsd:sequence>
               <xsd:element name="Book"  maxOccurs="unbounded" type="BookType"/>
           </xsd:sequence>
       </xsd:complexType>

       <xsd:complexType name="Book    ">
           <xsd:sequence>
               <xsd:element name="Title" type="xsd:string"/>
               <xsd:element name="Author" type="xsd:string"/>
               <xsd:element name "Date" type="xsd:string"/>
               <xsd:element name="   N" type="xsd:string"/>
               <xsd:element name="P      r" type="xsd:string"/>
           </xsd:sequence>
       </xsd:complexType>

</xsd:schema
```

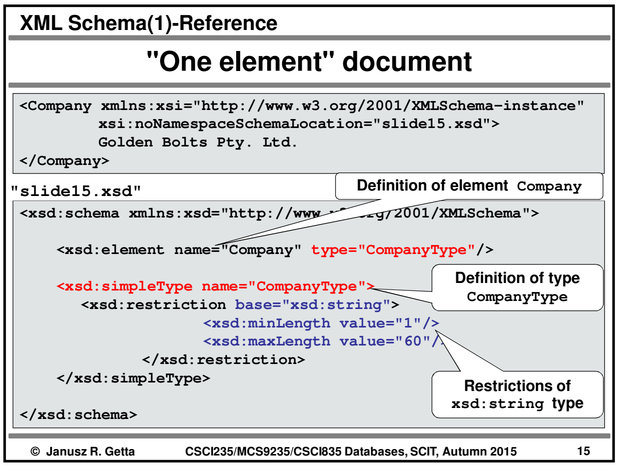
<!ELEMENT BookStore (Book+)>

<!ELEMENT Title (#PCDATA)>

<!ELEMENT Author (#PCDATA)>

<!ELEMENT Book (Title, Author, Date, ISBN, Publisher)>

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 14



XML Schema(1)-Reference

"One element" document

```
<Company xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="slide15.xsd">
         Golden Bolts Pty. Ltd.
</Company>
```

"slide15.xsd"

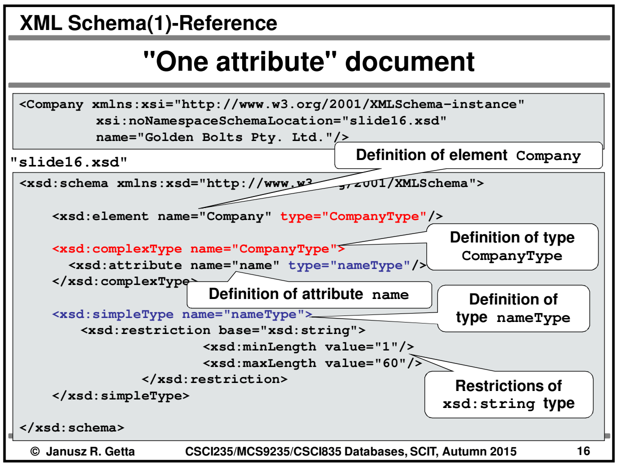
```
<xsd:schema xmlns:xsd="http://www     rg/2001/XMLSchema">

    <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>
```

Definition of element Company

Definition of type CompanyType

Restrictions of xsd:string type

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 15



XML Schema(1)-Reference

"One attribute" document

```
<Company xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="slide16.xsd"
         name="Golden Bolts Pty. Ltd."/>
```

"slide16.xsd"

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

    <xsd:element name="Company" type="CompanyType"/>

    <xsd:complexType name="CompanyType">
      <xsd:attribute name="name" type="nameType"/>
    </xsd:complexType>

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

</xsd:schema>
```

Definition of element Company

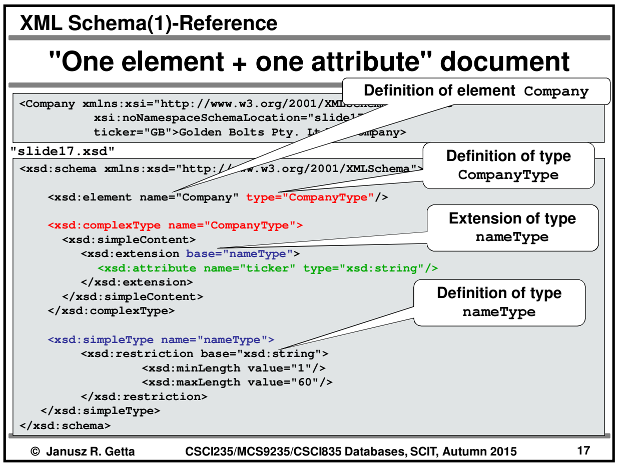
Definition of type CompanyType

Definition of attribute name

Definition of type nameType

Restrictions of xsd:string type

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 16



XML Schema(1)-Reference

"One element + one attribute" document

```
<Company xmlns:xsi="http://www.w3.org/2001/XML
          xsi:noNamespaceSchemaLocation="slide1
          ticker="GB">Golden Bolts Pty. Lt     ompany>
```

"slide17.xsd"

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

    <xsd:element name="Company" type="CompanyType"/>

    <xsd:complexType name="CompanyType">
      <xsd:simpleContent>
         <xsd:extension base="nameType">
           <xsd:attribute name="ticker" type="xsd:string"/>
         </xsd:extension>
      </xsd:simpleContent>
    </xsd:complexType>

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

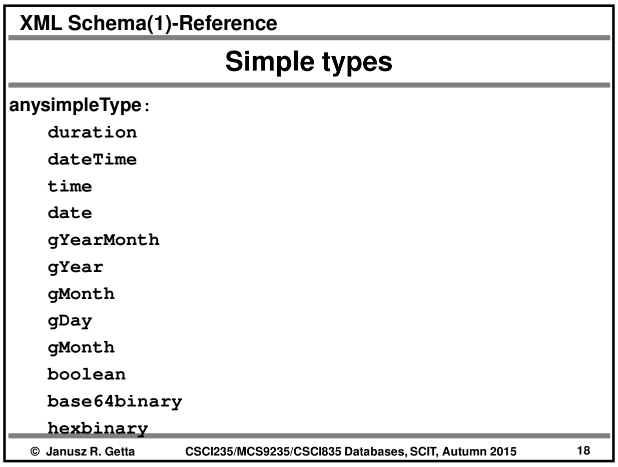
Definition of element Company

Definition of type CompanyType

Extension of type nameType

Definition of type nameType

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 17



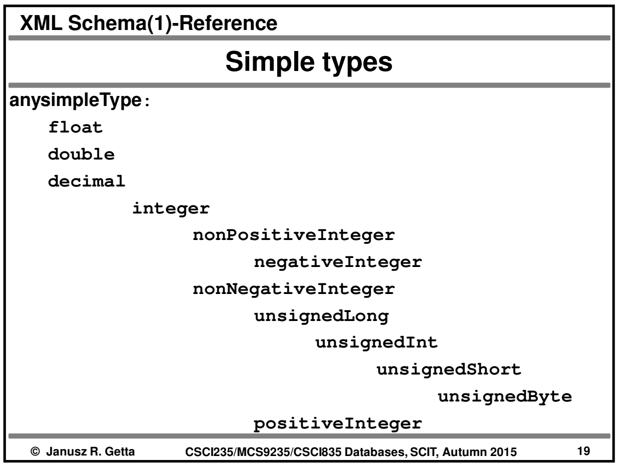
XML Schema(1)-Reference

Simple types

anysimpleType:

- `duration`
- `dateTime`
- `time`
- `date`
- `gYearMonth`
- `gYear`
- `gMonth`
- `gDay`
- `gMonth`
- `boolean`
- `base64binary`
- `hexbinary`

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 18



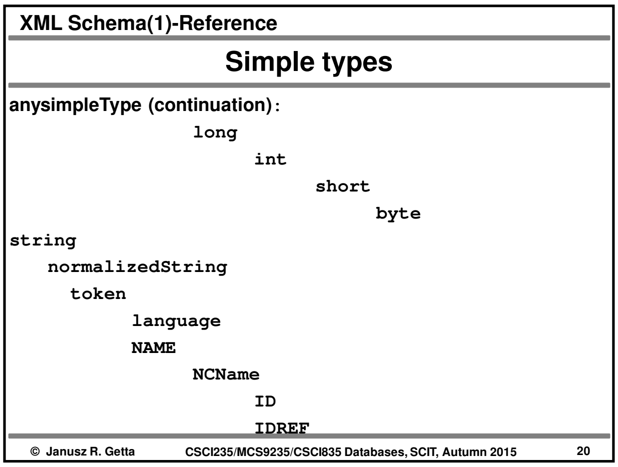
XML Schema(1)-Reference

Simple types

anysimpleType:

```
    float
    double
    decimal
           integer
                 nonPositiveInteger
                      negativeInteger
                 nonNegativeInteger
                      unsignedLong
                            unsignedInt
                                 unsignedShort
                                       unsignedByte
                      positiveInteger
```

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 19



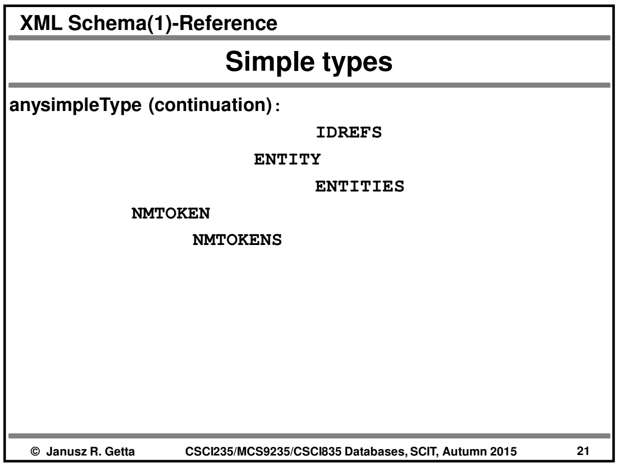
XML Schema(1)-Reference

Simple types

anysimpleType (continuation):

```
                 long
                      int
                            short
                                 byte
string
   normalizedString
     token
           language
           NAME
                 NCName
                      ID
                      IDREF
```

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 20



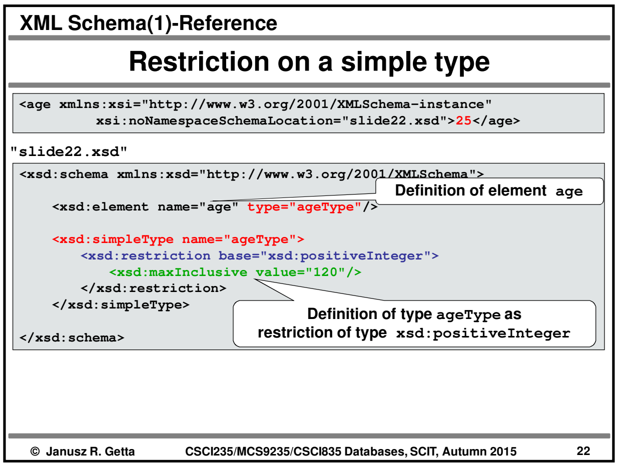
XML Schema(1)-Reference

Simple types

anysimpleType (continuation):

```
                            IDREFS
                      ENTITY
                            ENTITIES
           NMTOKEN
                 NMTOKENS
```

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 21



XML Schema(1)-Reference

Restriction on a simple type

```
<age xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="slide22.xsd">25</age>
```

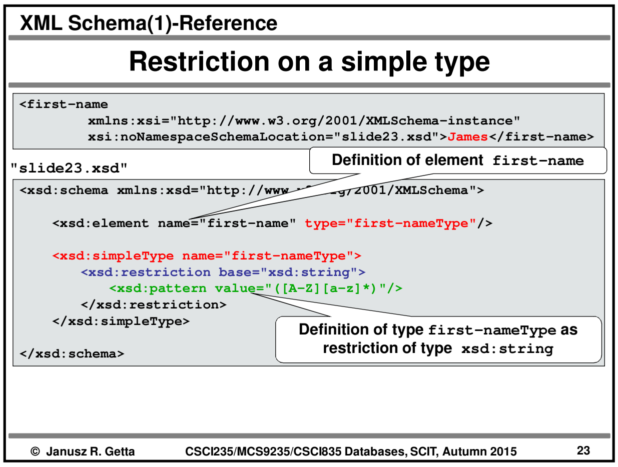
"slide22.xsd"

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

    <xsd:element name="age" type="ageType"/>

    <xsd:simpleType name="ageType">
        <xsd:restriction base="xsd:positiveInteger">
            <xsd:maxInclusive value="120"/>
        </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
```

Definition of element age

Definition of type ageType as restriction of type xsd:positiveInteger

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 22



XML Schema(1)-Reference

Restriction on a simple type

```
<first-name
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="slide23.xsd">James</first-name>
```

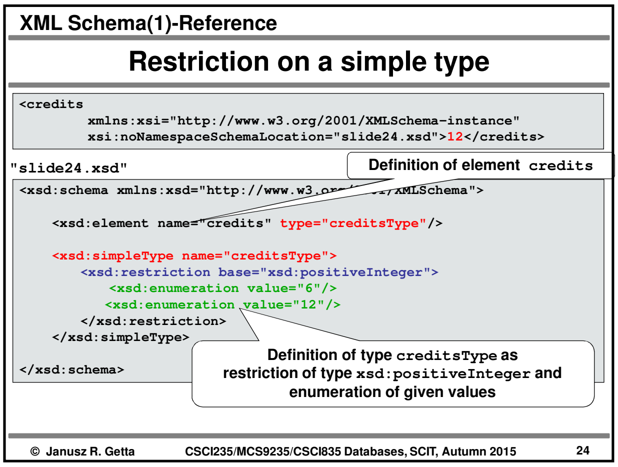
"slide23.xsd"

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

    <xsd:element name="first-name" type="first-nameType"/>

    <xsd:simpleType name="first-nameType">
        <xsd:restriction base="xsd:string">
            <xsd:pattern value="([A-Z][a-z]*)"/>
        </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
```

Definition of element first-name

Definition of type first-nameType as restriction of type xsd:string

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 23



XML Schema(1)-Reference

Restriction on a simple type

```
<credits
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:noNamespaceSchemaLocation="slide24.xsd">12</credits>
```

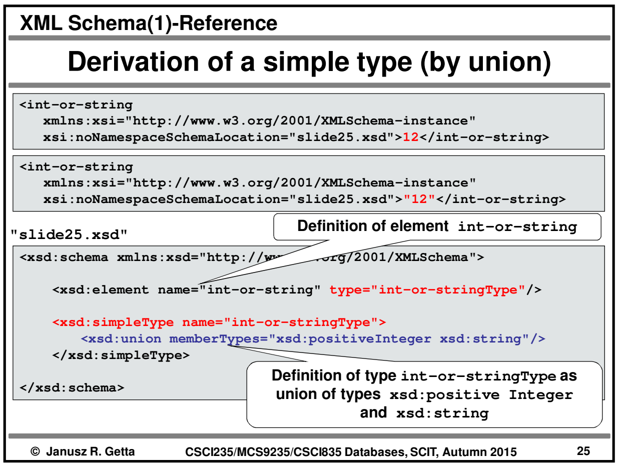
"slide24.xsd"

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

    <xsd:element name="credits" type="creditsType"/>

    <xsd:simpleType name="creditsType">
        <xsd:restriction base="xsd:positiveInteger">
            <xsd:enumeration value="6"/>
           <xsd:enumeration value="12"/>
        </xsd:restriction>
    </xsd:simpleType>

</xsd:schema>
```

Definition of element credits

Definition of type creditsType as restriction of type xsd:positiveInteger and enumeration of given values

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 24



XML Schema(1)-Reference

Derivation of a simple type (by union)

```
<int-or-string
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide25.xsd">12</int-or-string>
```

```
<int-or-string
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide25.xsd">"12"</int-or-string>
```

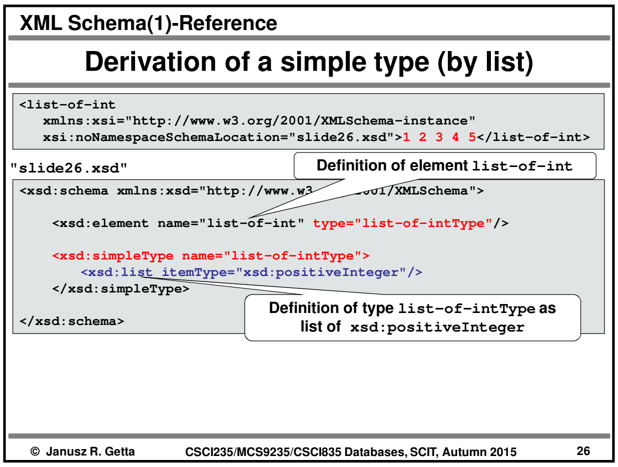
"slide25.xsd"

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

    <xsd:element name="int-or-string" type="int-or-stringType"/>

    <xsd:simpleType name="int-or-stringType">
       <xsd:union memberTypes="xsd:positiveInteger xsd:string"/>
    </xsd:simpleType>

</xsd:schema>
```

Definition of element int-or-string

Definition of type int-or-stringType as union of types xsd:positive Integer and xsd:string

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 25



XML Schema(1)-Reference

Derivation of a simple type (by list)

```
<list-of-int
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide26.xsd">1 2 3 4 5</list-of-int>
```

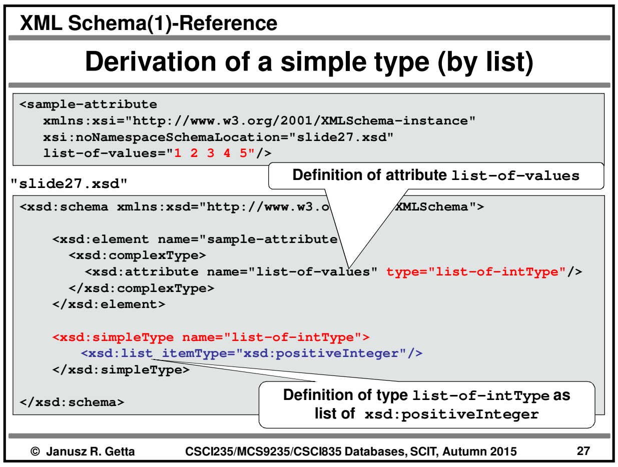
"slide26.xsd"

```
<xsd:schema xmlns:xsd="http://www.w3      001/XMLSchema">

    <xsd:element name="list-of-int" type="list-of-intType"/>

    <xsd:simpleType name="list-of-intType">
       <xsd:list itemType="xsd:positiveInteger"/>
    </xsd:simpleType>

</xsd:schema>
```

Definition of element list-of-int

Definition of type list-of-intType as list of xsd:positiveInteger

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 26



XML Schema(1)-Reference

Derivation of a simple type (by list)

```
<sample-attribute
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide27.xsd"
   list-of-values="1 2 3 4 5"/>
```

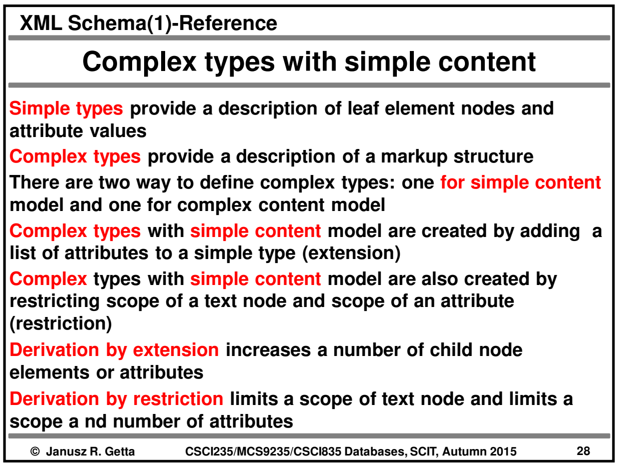
"slide27.xsd"

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

    <xsd:element name="sample-attribute
      <xsd:complexType>
        <xsd:attribute name="list-of-values" type="list-of-intType"/>
      </xsd:complexType>
    </xsd:element>

    <xsd:simpleType name="list-of-intType">
       <xsd:list itemType="xsd:positiveInteger"/>
    </xsd:simpleType>

</xsd:schema>
```

Definition of attribute list-of-values

Definition of type list-of-intType as list of xsd:positiveInteger

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 27



XML Schema(1)-Reference

Complex types with simple content

Simple types provide a description of leaf element nodes and attribute values

Complex types provide a description of a markup structure

There are two way to define complex types: one for simple content model and one for complex content model

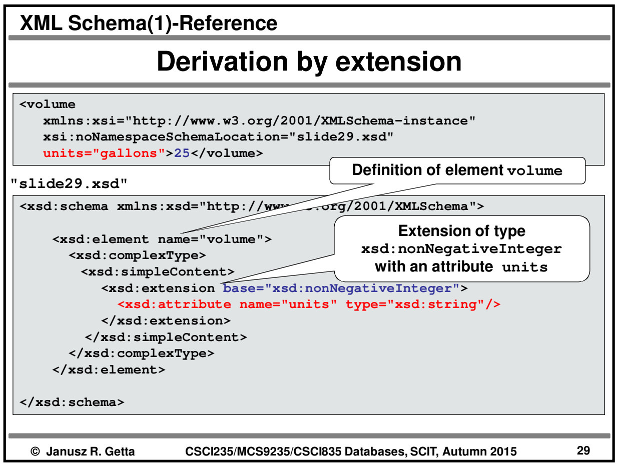
Complex types with simple content model are created by adding a list of attributes to a simple type (extension)

Complex types with simple content model are also created by restricting scope of a text node and scope of an attribute (restriction)

Derivation by extension increases a number of child node elements or attributes

Derivation by restriction limits a scope of text node and limits a scope a nd number of attributes

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 28



XML Schema(1)-Reference

Derivation by extension

```
<volume
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide29.xsd"
   units="gallons">25</volume>
```

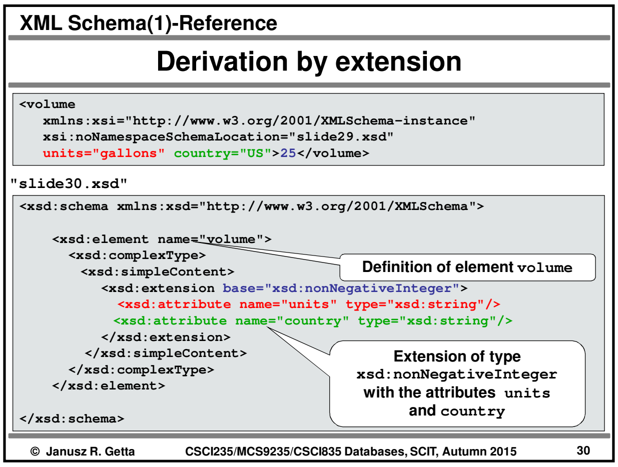
"slide29.xsd"

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

    <xsd:element name="volume">
      <xsd:complexType>
        <xsd:simpleContent>
           <xsd:extension base="xsd:nonNegativeInteger">
             <xsd:attribute name="units" type="xsd:string"/>
           </xsd:extension>
         </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>

</xsd:schema>
```

Definition of element volume

Extension of type xsd:nonNegativeInteger with an attribute units

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 29



XML Schema(1)-Reference

Derivation by extension

```
<volume
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide29.xsd"
   units="gallons" country="US">25</volume>
```

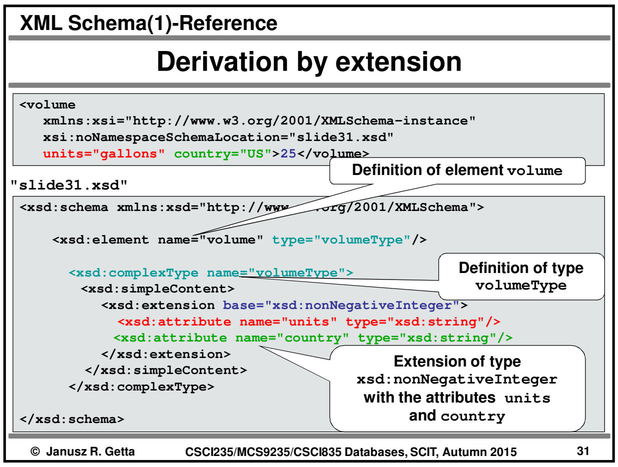
"slide30.xsd"

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

    <xsd:element name="volume">
      <xsd:complexType>
        <xsd:simpleContent>
           <xsd:extension base="xsd:nonNegativeInteger">
             <xsd:attribute name="units" type="xsd:string"/>
            <xsd:attribute name="country" type="xsd:string"/>
           </xsd:extension>
         </xsd:simpleContent>
      </xsd:complexType>
    </xsd:element>

</xsd:schema>
```

Definition of element volume

Extension of type xsd:nonNegativeInteger with the attributes units and country

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 30



XML Schema(1)-Reference

Derivation by extension

```
<volume
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide31.xsd"
   units="gallons" country="US">25</volume>
```

"slide31.xsd"

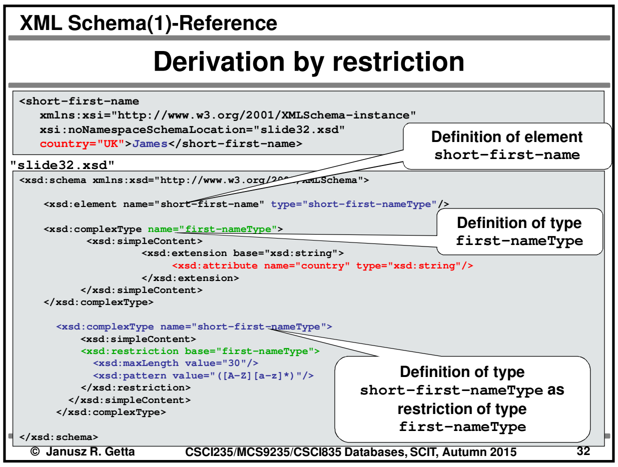
```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <xsd:element name="volume" type="volumeType"/>

      <xsd:complexType name="volumeType">
        <xsd:simpleContent>
          <xsd:extension base="xsd:nonNegativeInteger">
            <xsd:attribute name="units" type="xsd:string"/>
            <xsd:attribute name="country" type="xsd:string"/>
          </xsd:extension>
        </xsd:simpleContent>
      </xsd:complexType>

</xsd:schema>
```

Definition of element `volume`

Definition of type `volumeType`

Extension of type `xsd:nonNegativeInteger` with the attributes `units` and `country`

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 31



XML Schema(1)-Reference

Derivation by restriction

```
<short-first-name
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:noNamespaceSchemaLocation="slide32.xsd"
   country="UK">James</short-first-name>
```

"slide32.xsd"

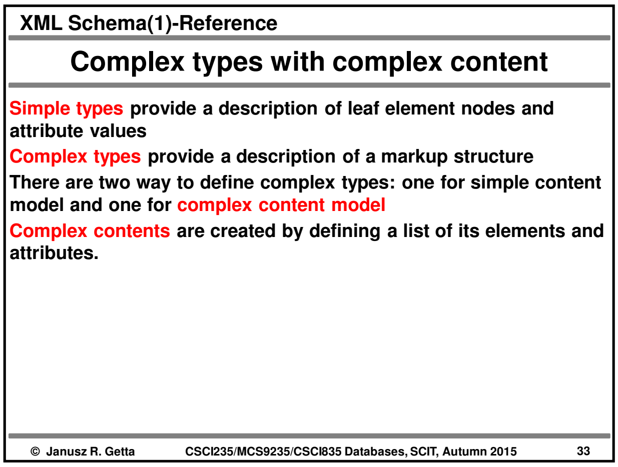
```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

    <xsd:element name="short-first-name" type="short-first-nameType"/>

    <xsd:complexType name="first-nameType">
           <xsd:simpleContent>
                   <xsd:extension base="xsd:string">
                        <xsd:attribute name="country" type="xsd:string"/>
                   </xsd:extension>
          </xsd:simpleContent>
    </xsd:complexType>

      <xsd:complexType name="short-first-nameType">
          <xsd:simpleContent>
          <xsd:restriction base="first-nameType">
            <xsd:maxLength value="30"/>
            <xsd:pattern value="([A-Z][a-z]*)"/>
          </xsd:restriction>
        </xsd:simpleContent>
      </xsd:complexType>

</xsd:schema>
```

Definition of element `short-first-name`

Definition of type `first-nameType`

Definition of type `short-first-nameType` as restriction of type `first-nameType`

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 32



XML Schema(1)-Reference

Complex types with complex content

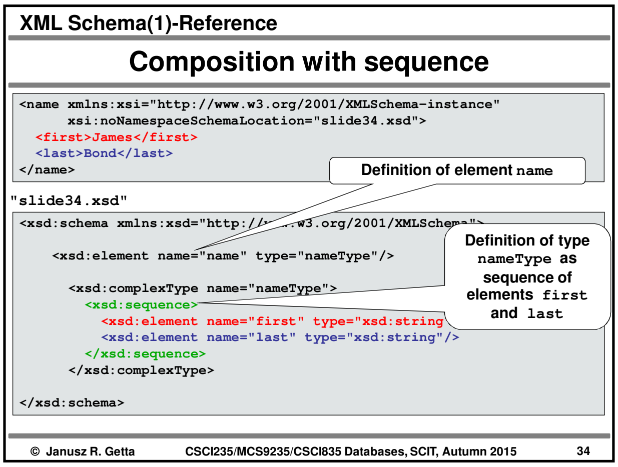
Simple types provide a description of leaf element nodes and attribute values

Complex types provide a description of a markup structure

There are two way to define complex types: one for simple content model and one for complex content model

Complex contents are created by defining a list of its elements and attributes.

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 33



XML Schema(1)-Reference

Composition with sequence

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide34.xsd">
  <first>James</first>
  <last>Bond</last>
</name>
```

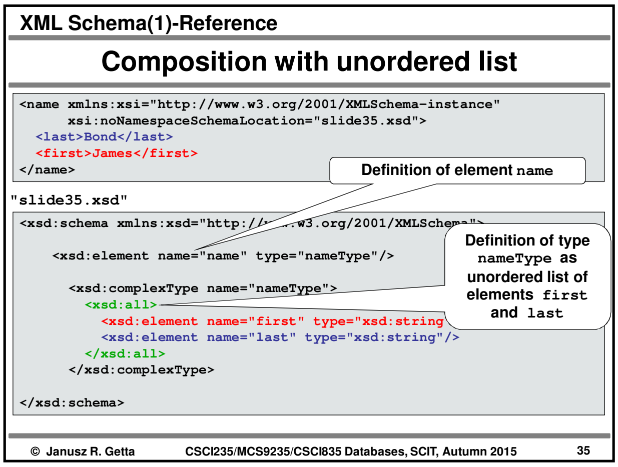
"slide34.xsd"

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

    <xsd:element name="name" type="nameType"/>

      <xsd:complexType name="nameType">
        <xsd:sequence>
          <xsd:element name="first" type="xsd:string
          <xsd:element name="last" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>

</xsd:schema>
```

Definition of element `name`

Definition of type `nameType` as sequence of elements `first` and `last`

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 34



XML Schema(1)-Reference

Composition with unordered list

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide35.xsd">
  <last>Bond</last>
  <first>James</first>
</name>
```

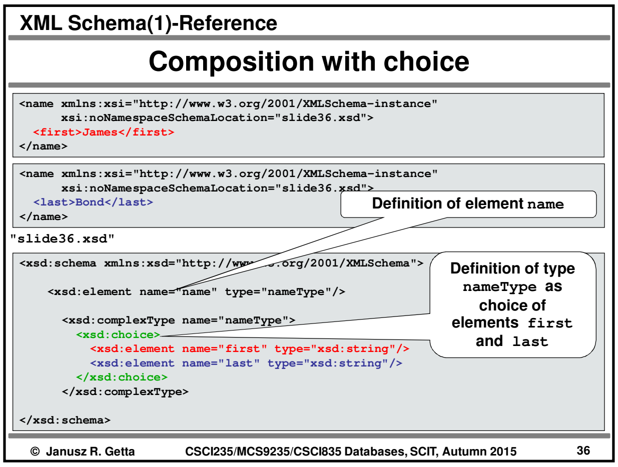
"slide35.xsd"

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

    <xsd:element name="name" type="nameType"/>

      <xsd:complexType name="nameType">
        <xsd:all>
          <xsd:element name="first" type="xsd:string
          <xsd:element name="last" type="xsd:string"/>
        </xsd:all>
      </xsd:complexType>

</xsd:schema>
```

Definition of element `name`

Definition of type `nameType` as unordered list of elements `first` and `last`

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 35



XML Schema(1)-Reference

Composition with choice

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide36.xsd">
  <first>James</first>
</name>
```

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide36.xsd">
  <last>Bond</last>
</name>
```

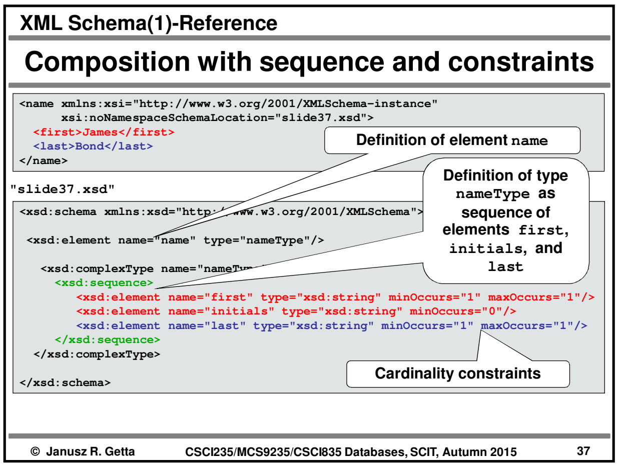
"slide36.xsd"

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

    <xsd:element name="name" type="nameType"/>

      <xsd:complexType name="nameType">
        <xsd:choice>
          <xsd:element name="first" type="xsd:string"/>
          <xsd:element name="last" type="xsd:string"/>
        </xsd:choice>
      </xsd:complexType>

</xsd:schema>
```

Definition of element `name`

Definition of type `nameType` as choice of elements `first` and `last`

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 36



XML Schema(1)-Reference

Composition with sequence and constraints

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide37.xsd">
  <first>James</first>
  <last>Bond</last>
</name>
```

"slide37.xsd"

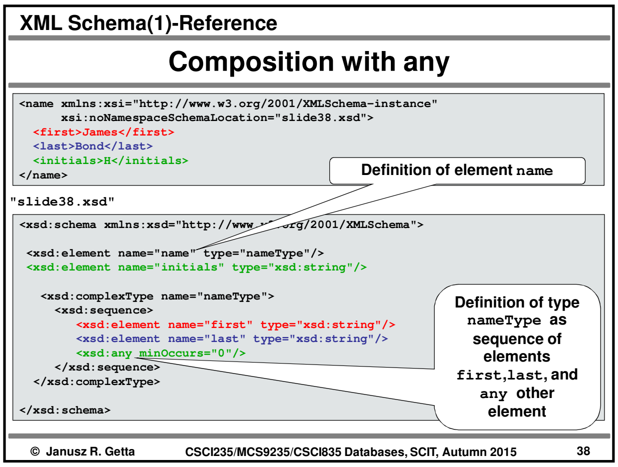
```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

 <xsd:element name="name" type="nameType"/>

   <xsd:complexType name="nameType">
     <xsd:sequence>
        <xsd:element name="first" type="xsd:string" minOccurs="1" maxOccurs="1"/>
        <xsd:element name="initials" type="xsd:string" minOccurs="0"/>
        <xsd:element name="last" type="xsd:string" minOccurs="1" maxOccurs="1"/>
     </xsd:sequence>
  </xsd:complexType>

</xsd:schema>
```

Definition of element `name`

Definition of type `nameType` as sequence of elements `first`, `initials`, and `last`

Cardinality constraints

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 37



XML Schema(1)-Reference

Composition with any

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide38.xsd">
  <first>James</first>
  <last>Bond</last>
  <initials>H</initials>
</name>
```

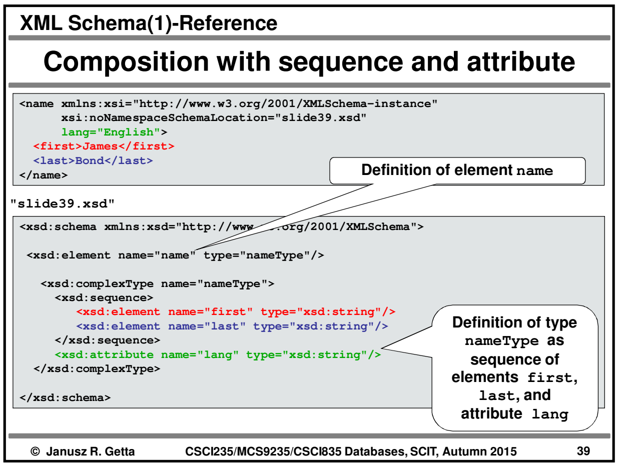
"slide38.xsd"

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

 <xsd:element name="name" type="nameType"/>
 <xsd:element name="initials" type="xsd:string"/>

   <xsd:complexType name="nameType">
     <xsd:sequence>
        <xsd:element name="first" type="xsd:string"/>
        <xsd:element name="last" type="xsd:string"/>
        <xsd:any minOccurs="0"/>
     </xsd:sequence>
  </xsd:complexType>

</xsd:schema>
```

Definition of element `name`

Definition of type `nameType` as sequence of elements `first`,`last`, and `any` other element

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 38



XML Schema(1)-Reference

Composition with sequence and attribute

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide39.xsd"
      lang="English">
  <first>James</first>
  <last>Bond</last>
</name>
```

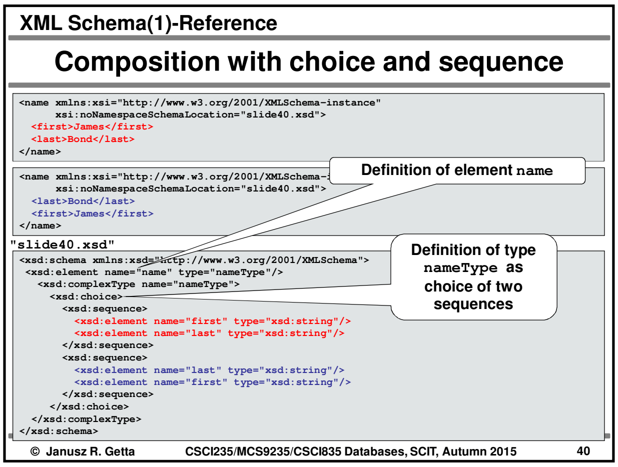
"slide39.xsd"

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

 <xsd:element name="name" type="nameType"/>

   <xsd:complexType name="nameType">
     <xsd:sequence>
        <xsd:element name="first" type="xsd:string"/>
        <xsd:element name="last" type="xsd:string"/>
     </xsd:sequence>
     <xsd:attribute name="lang" type="xsd:string"/>
  </xsd:complexType>

</xsd:schema>
```

Definition of element `name`

Definition of type `nameType` as sequence of elements `first`, `last`, and attribute `lang`

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 39



XML Schema(1)-Reference

Composition with choice and sequence

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide40.xsd">
  <first>James</first>
  <last>Bond</last>
</name>
```

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide40.xsd">
  <last>Bond</last>
  <first>James</first>
</name>
```

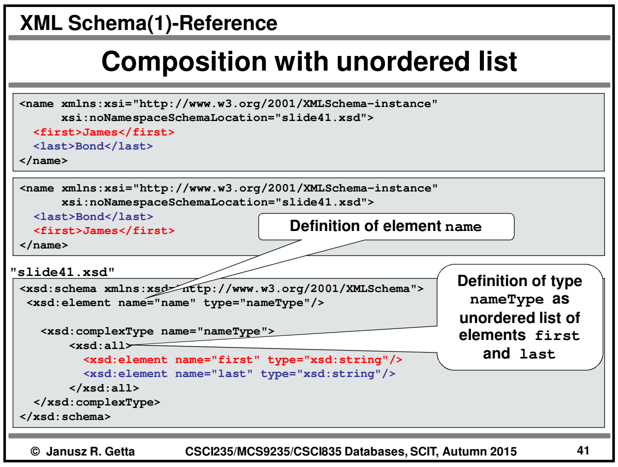
"slide40.xsd"

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="name" type="nameType"/>
   <xsd:complexType name="nameType">
     <xsd:choice>
       <xsd:sequence>
         <xsd:element name="first" type="xsd:string"/>
         <xsd:element name="last" type="xsd:string"/>
       </xsd:sequence>
       <xsd:sequence>
         <xsd:element name="last" type="xsd:string"/>
         <xsd:element name="first" type="xsd:string"/>
       </xsd:sequence>
     </xsd:choice>
  </xsd:complexType>
</xsd:schema>
```

Definition of element `name`

Definition of type `nameType` as choice of two sequences

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 40



XML Schema(1)-Reference

Composition with unordered list

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide41.xsd">
  <first>James</first>
  <last>Bond</last>
</name>
```

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide41.xsd">
  <last>Bond</last>
  <first>James</first>
</name>
```

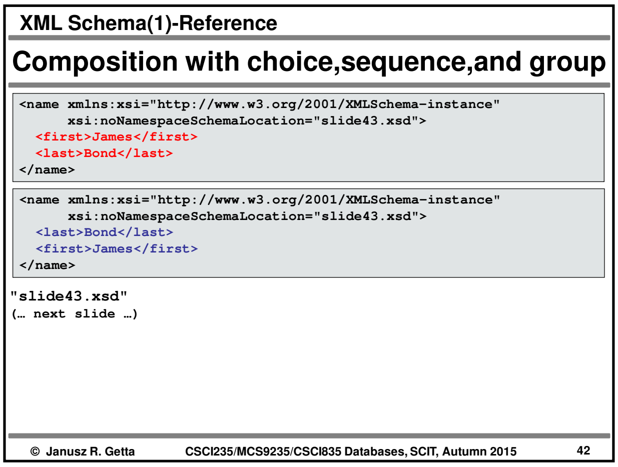
"slide41.xsd"

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

   <xsd:complexType name="nameType">
       <xsd:all>
         <xsd:element name="first" type="xsd:string"/>
         <xsd:element name="last" type="xsd:string"/>
       </xsd:all>
  </xsd:complexType>
</xsd:schema>
```

Definition of element `name`

Definition of type `nameType` as unordered list of elements `first` and `last`

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 41



XML Schema(1)-Reference

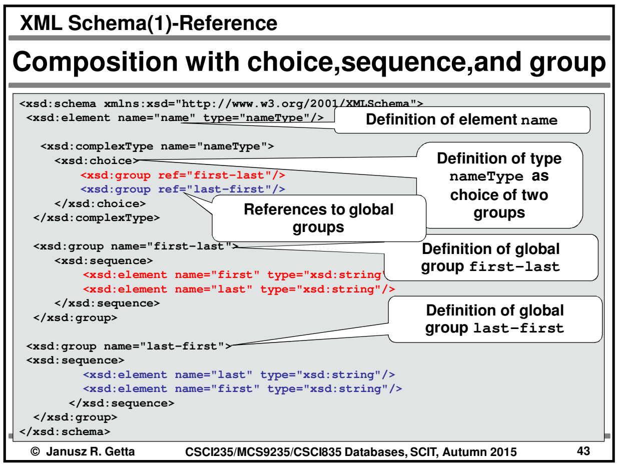
Composition with choice,sequence,and group

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide43.xsd">
  <first>James</first>
  <last>Bond</last>
</name>
```

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide43.xsd">
  <last>Bond</last>
  <first>James</first>
</name>
```

"slide43.xsd"
(… next slide …)

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 42



XML Schema(1)-Reference

Composition with choice,sequence,and group

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

   <xsd:complexType name="nameType">
     <xsd:choice>
         <xsd:group ref="first-last"/>
         <xsd:group ref="last-first"/>
     </xsd:choice>
  </xsd:complexType>

  <xsd:group name="first-last">
     <xsd:sequence>
         <xsd:element name="first" type="xsd:string
         <xsd:element name="last" type="xsd:string"/>
     </xsd:sequence>
  </xsd:group>

 <xsd:group name="last-first">
 <xsd:sequence>
         <xsd:element name="last" type="xsd:string"/>
         <xsd:element name="first" type="xsd:string"/>
       </xsd:sequence>
  </xsd:group>
</xsd:schema>
```

Definition of element `name`

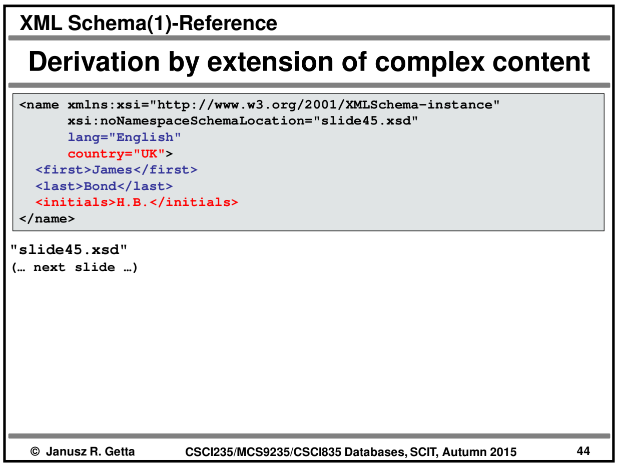
Definition of type `nameType` as choice of two groups

References to global groups

Definition of global group `first-last`

Definition of global group `last-first`

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 43



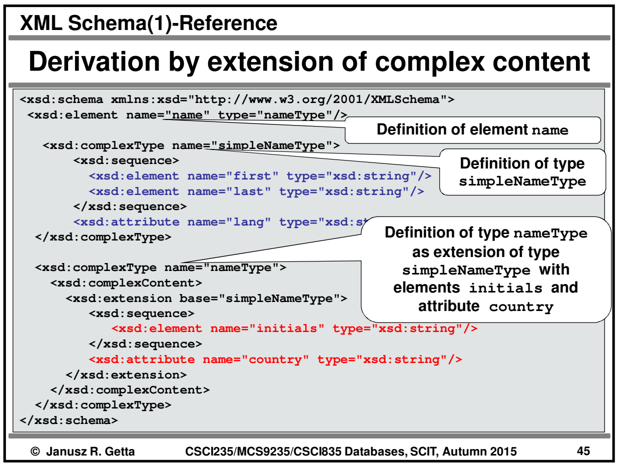
XML Schema(1)-Reference

Derivation by extension of complex content

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide45.xsd"
      lang="English"
      country="UK">
  <first>James</first>
  <last>Bond</last>
  <initials>H.B.</initials>
</name>
```

"slide45.xsd"
(… next slide …)

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 44



XML Schema(1)-Reference

Derivation by extension of complex content

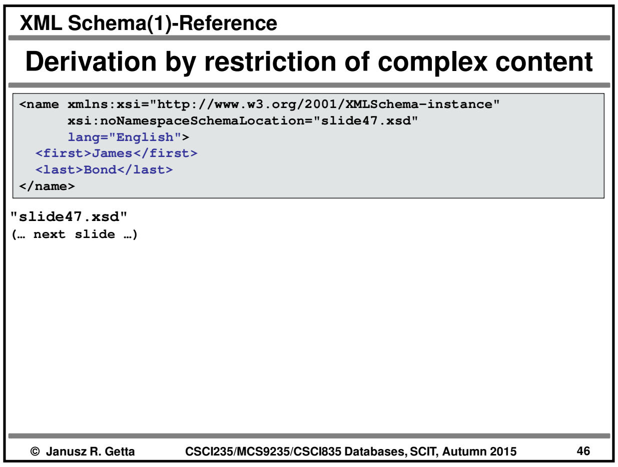
```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="name" type="nameType"/>

   <xsd:complexType name="simpleNameType">
       <xsd:sequence>
         <xsd:element name="first" type="xsd:string"/>
         <xsd:element name="last" type="xsd:string"/>
       </xsd:sequence>
       <xsd:attribute name="lang" type="xsd:st
  </xsd:complexType>

  <xsd:complexType name="nameType">
    <xsd:complexContent>
      <xsd:extension base="simpleNameType">
         <xsd:sequence>
            <xsd:element name="initials" type="xsd:string"/>
         </xsd:sequence>
         <xsd:attribute name="country" type="xsd:string"/>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

Definition of element `name`

Definition of type `simpleNameType`

Definition of type `nameType` as extension of type `simpleNameType` with elements `initials` and attribute `country`

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 45



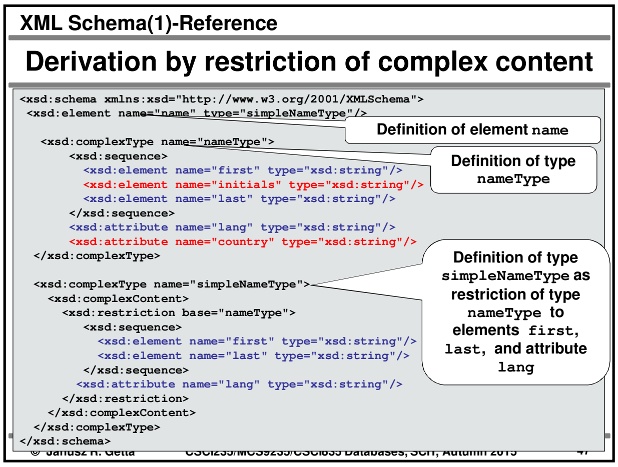
XML Schema(1)-Reference

Derivation by restriction of complex content

```
<name xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide47.xsd"
      lang="English">
  <first>James</first>
  <last>Bond</last>
</name>
```

"slide47.xsd"
(… next slide …)

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 46



XML Schema(1)-Reference

Derivation by restriction of complex content

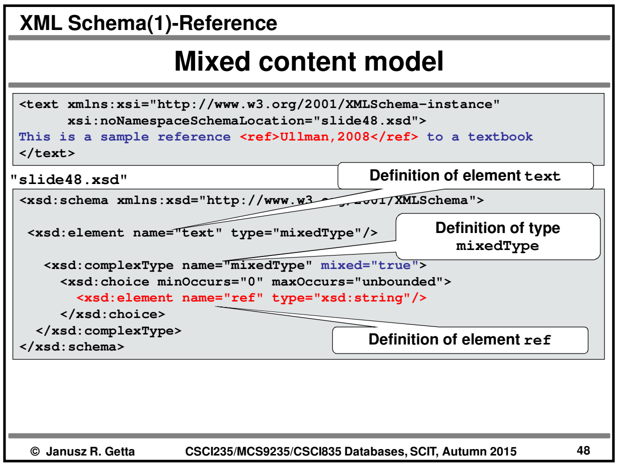
```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <xsd:element name="name" type="simpleNameType"/>

   <xsd:complexType name="nameType">
       <xsd:sequence>
         <xsd:element name="first" type="xsd:string"/>
         <xsd:element name="initials" type="xsd:string"/>
         <xsd:element name="last" type="xsd:string"/>
       </xsd:sequence>
       <xsd:attribute name="lang" type="xsd:string"/>
       <xsd:attribute name="country" type="xsd:string"/>
  </xsd:complexType>

  <xsd:complexType name="simpleNameType">
    <xsd:complexContent>
      <xsd:restriction base="nameType">
         <xsd:sequence>
           <xsd:element name="first" type="xsd:string"/>
           <xsd:element name="last" type="xsd:string"/>
         </xsd:sequence>
        <xsd:attribute name="lang" type="xsd:string"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

Definition of element `name`

Definition of type `nameType`

Definition of type `simpleNameType` as restriction of type `nameType` to elements `first`, `last`, and attribute `lang`

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 47



XML Schema(1)-Reference

Mixed content model

```
<text xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide48.xsd">
This is a sample reference <ref>Ullman,2008</ref> to a textbook
</text>
```

"slide48.xsd"

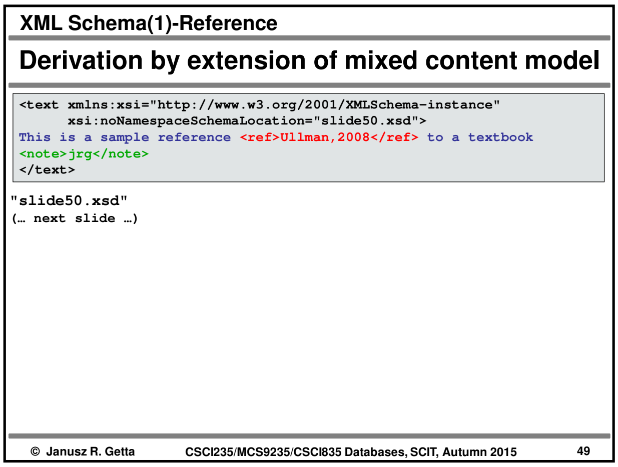
```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

 <xsd:element name="text" type="mixedType"/>

   <xsd:complexType name="mixedType" mixed="true">
     <xsd:choice minOccurs="0" maxOccurs="unbounded">
       <xsd:element name="ref" type="xsd:string"/>
     </xsd:choice>
   </xsd:complexType>
</xsd:schema>
```

Definition of element `text`

Definition of type `mixedType`

Definition of element `ref`

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 48



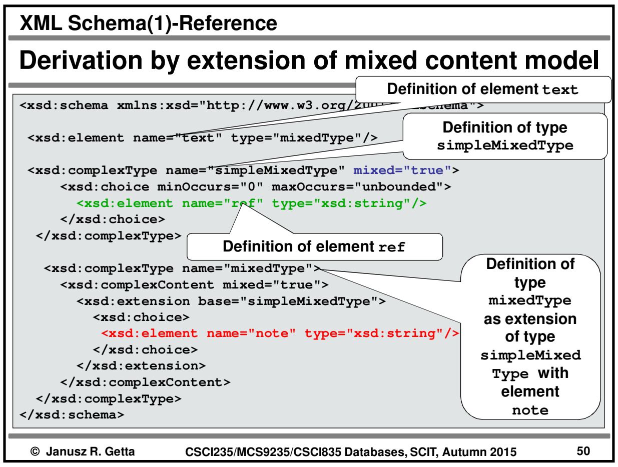
XML Schema(1)-Reference

Derivation by extension of mixed content model

```
<text xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="slide50.xsd">
This is a sample reference <ref>Ullman,2008</ref> to a textbook
<note>jrg</note>
</text>
```

"slide50.xsd"
(… next slide …)

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 49



XML Schema(1)-Reference

Derivation by extension of mixed content model

```
<xsd:schema xmlns:xsd="http://www.w3.org/20[illegible]schema">

 <xsd:element name="text" type="mixedType"/>

 <xsd:complexType name="simpleMixedType" mixed="true">
     <xsd:choice minOccurs="0" maxOccurs="unbounded">
       <xsd:element name="ref" type="xsd:string"/>
     </xsd:choice>
  </xsd:complexType>

   <xsd:complexType name="mixedType">
     <xsd:complexContent mixed="true">
       <xsd:extension base="simpleMixedType">
         <xsd:choice>
          <xsd:element name="note" type="xsd:string"/>
         </xsd:choice>
       </xsd:extension>
     </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

Definition of element `text`

Definition of type `simpleMixedType`

Definition of element `ref`

Definition of type `mixedType` as extension of type `simpleMixedType` with element `note`

© Janusz R. Getta CSCI235/MCS9235/CSCI835 Databases, SCIT, Autumn 2015 50