

XML & Databases

연구2그룹 신기술 세미나



Young-il Cho

DB Solution Team

2005/09/24

❖ Contents

1. Introduction

2. New Features

3. Solutions in Market

4. Conclusion

5. Reference



Introduction

❖ Introduction

- XML
- DTD



❖ What is XML?

- **EX**tensible **M**arkup **L**anguage
- Markup language just like HTML
- Designed to describe data
 - You must define you own tags
- Self-Descriptive
 - Use DTD or XML Schema
- W3C recommendation (1998)
 - Current state: XML 1.0, 3rd ed. (2004)
 - Last draft: XML 1.1 (2004)

❖ DTD

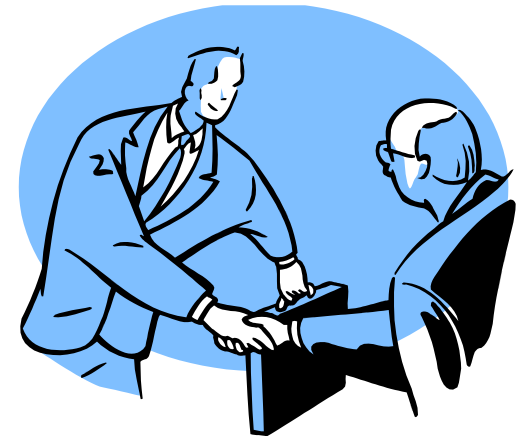
- Document Type Definition
 - Internal DOCTYPE

```
<?xml version="1.0"?>
<!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend!</body>
</note>
```



❖ DTD Features

- External DOCTYPE declaration
 - `<!DOCTYPE root-element SYSTEM "filename">`
 - `<!DOCTYPE note SYSTEM "note.dtd">`
- People can agree to use a common DTD for interchanging data
- Use Standard DTD
 - For document exchange
 - Save DTD design works
 - Examples
 - MathML, CML, UXF, SMIL, RDF, HumanML, DocBook





❖ DTD Structure

- Elements: main building blocks
- Tags: mark up elements
 - `<element> contents </element>`
- Attributes: extra information about elements
- Entities: common text
 - `<`, `>`, `&`, `"`, `'`, ...
- PCDATA: parsed character data
- CDATA: character data (will not be expanded)



DTD - Introduction

❖ Example

```
<!DOCTYPE NEWSPAPER [  
<!ELEMENT NEWSPAPER (ARTICLE+)>  
<!ELEMENT ARTICLE (HEADLINE,BYLINE,LEAD,BODY,NOTES)>  
<!ELEMENT HEADLINE (#PCDATA)>  
<!ELEMENT BYLINE (#PCDATA)>  
<!ELEMENT LEAD (#PCDATA)>  
<!ELEMENT BODY (#PCDATA)>  
<!ELEMENT NOTES (#PCDATA)>  
<!ATTLIST ARTICLE AUTHOR CDATA #REQUIRED>  
<!ATTLIST ARTICLE EDITOR CDATA #IMPLIED>  
<!ATTLIST ARTICLE DATE CDATA #IMPLIED>  
<!ATTLIST ARTICLE EDITION CDATA #IMPLIED>  
<!ENTITY NEWSPAPER "Vervet Logic Times">  
<!ENTITY PUBLISHER "Vervet Logic Press">  
<!ENTITY COPYRIGHT "Copyright 1998 Vervet Logic Press">  
>
```




New Features

❖ New Features

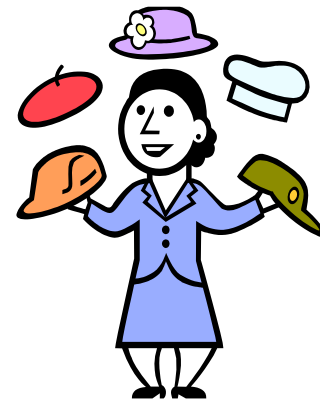
- Recent Trends
- XML Schema
- XPath
- XQuery
- SQL/XML



Recent Trends - New Features

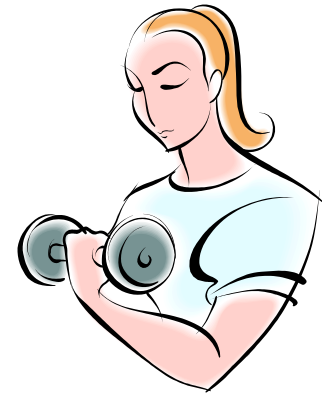
❖ Recent Trends

- Evolution of XML programming model
 - SAX (Simple API for XML)
 - DOM (Document Object Model)
 - XSLT (Extensible Stylesheet Language Transformation)
 - * SOAP (Simple Object Access Protocol)
- Language approach
 - XQuery
 - SQL/XML
 - XSLT & XPath



❖ XML Schema

- An alternative to DTD written in XML
- Extensible for future additions
- Richer and more useful than DTD
- Proper data types & namespaces
- XML Schema is converted to DB Schema easily
- W3C Recommendation





XML Schema - New Features

❖ Easy Example

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.w3schools.com"
  xmlns="http://www.w3schools.com" elementFormDefault="qualified">
<xs:element name="note">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```



XML Schema - New Features

❖ XML Schema : Simple Types

– Element

- `<xs:element name="age" type="integer"/>`

– Attribute

- `<xs:attribute name="lang" type="xs:string"/>`

– Restriction: control acceptable values

- `<xs:minInclusive value="0"/>`
- `<xs:enumeration value="Audi"/>`
- `<xs:pattern value="[A-Z][a-z]*"/>`
- `<xs:maxLength value="8"/>`

❖ XML Schema : Complex Types

– Elements

```
<xs:element name="employee">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="firstname" type="xs:string"/>  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```

▪ Empty

- ``, `<product prodid="1345"/>`
- Do not declare any elements

▪ Elements only

- `<xs:complexType>` and `<xs:sequence>`



XML Schema - New Features

❖ Example: XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<shiporder orderid="889923" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="shiporder.xsd">
  <orderperson>John Smith</orderperson>
  <shipto>
    <name>Ola Nordmann</name>   <address>Langgt 23</address>
    <city>4000 Stavanger</city>   <country>Norway</country>
  </shipto>
  <item>
    <title>Empire Burlesque</title>   <note>Special Edition</note>
    <quantity>1</quantity>           <price>10.90</price>
  </item>
  <item>
    <title>Hide your heart</title>   <quantity>1</quantity>   <price>9.90</price>
  </item>
</shiporder>
```



XML Schema - New Features

❖ XML Schema : Complex Types

- Elements
 - Text Only
 - `<xs:simpleContent>`
 - Mixed
 - `<xs:complexType ... mixed="true">`
 - Indicator
 - `<xs:all>`, `<xs:choice>`, `<xs:sequence>`,
`<xs:group ...>`, `<xs:attributeGroup ...>`,
`minOccurs`, `maxOccurs`
 - `any`, `anyAttribute`
 - `<xs:any ...>`, `<xs:anyAttribute/>`



XML Schema - New Features

❖ Example: Schema

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
  <xs:simpleType name="stringtype">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:simpleType name="inttype">
    <xs:restriction base="xs:positiveInteger"/>
  </xs:simpleType>
  <xs:simpleType name="dectype">
    <xs:restriction base="xs:decimal"/>
  </xs:simpleType>
  <xs:simpleType name="orderidtype">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]{6}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="shiptotype">
    <xs:sequence>
      <xs:element name="name" type="stringtype"/>
      <xs:element name="address"
        type="stringtype"/>
      <xs:element name="city" type="stringtype"/>

```

```
      <xs:element name="country" type="stringtype"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="itemtype">
    <xs:sequence>
      <xs:element name="title" type="stringtype"/>
      <xs:element name="note" type="stringtype"
        minOccurs="0"/>
      <xs:element name="quantity" type="inttype"/>
      <xs:element name="price" type="dectype"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="shipordertype">
    <xs:sequence>
      <xs:element name="orderperson" type="stringtype"/>
      <xs:element name="shipto" type="shiptotype"/>
      <xs:element name="item" maxOccurs="unbounded"
        type="itemtype"/>
    </xs:sequence>
    <xs:attribute name="orderid" type="orderidtype"
      use="required"/>
  </xs:complexType>
  <xs:element name="shiporder" type="shipordertype"/>
</xs:schema>
```



XPath - New Features

❖ XPath

- Syntax for defining parts of an XML document
- Uses path expression to navigate in XML documents
- Contains a library of standard functions
- Major element in XSLT
- W3C Standard

❖ XPath Structure

- Nodes: element, attribute, text, namespace, processing-instruction, comment, document node
- Relationship: parent, children, siblings, ancestors, descendants
- Operators and functions

❖ XPath Syntax

Expression	Description
nodename	All child nodes of the node
/	From the root node
//	Anywhere in the document
.	Current node
..	Parent of the current
@	Attributes
[]	Predicates
*	All
node()	Any nodes of any kind
	Or

❖ XQuery

- Same as ‘XML Query’
- Query language
- XQuery : XML = SQL : DB tables
- Built on XPath expressions
- Supported by all the major database engines
- W3C working draft
 - Inter-compatible among different recommendations
 - XQuery 1.0 & XPath 2.0 & XSLT 2.0





XQuery - New Features

❖ Example: XML

```
<?xml version="1.0"encoding="ISO-8859-1"?>
<bookstore>
<book category="COOKING">
<title lang="en">Everyday Italian</title>
<author>Giada De Laurentiis</author>
<year>2005</year>
<price>30.00</price>
</book>
<book category="CHILDREN">
<title lang="en">Harry Potter</title>
<author>J K. Rowling</author>
<year>2005</year>
<price>29.99</price>
</book>
<book category="WEB">
```

```
<title lang="en">XQuery Kick Start</title>
<author>James McGovern</author>
<author>Per Bothner</author>
<author>Kurt Cagle</author>
<author>James Linn</author>
<author>Vaidyanathan Nagarajan</author>
<year>2003</year> <price>49.99</price>
</book>
<book category="WEB">
<title lang="en">Learning XML</title>
<author>Erik T. Ray</author>
<year>2003</year>
<price>39.95</price>
</book>
</bookstore>
```

❖ Example: XQuery in XPath

- `doc("books.xml")/bookstore/book/title →`

```
<title lang="en">Everyday Italian</title>
<title lang="en">Harry Potter</title>
<title lang="en">XQuery Kick Start</title>
<title lang="en">Learning XML</title>
```

- `doc("books.xml")/bookstore/book[price<30] →`

```
<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
```

❖ FLWOR Expressions

- Use FLWOR expression instead of path expression
 - `doc("books.xml")/bookstore/book[price>30]/title`

```
for $x in doc("books.xml")/bookstore/book
where $x/price>30
order by $x/title
return $x/title
```

```
<title lang="en">Learning XML</title>
<title lang="en">XQuery Kick Start</title>
```

❖ XQuery in HTML

```
<ul>
{
for $x in doc("books.xml")/bookstore/book/title
order by $x
return <li>{data($x)}</li>
}
</ul>
```

```
<ul>
<li>Everyday Italian</li>
<li>Harry Potter</li>
<li>Learning XML</li>
<li>XQuery Kick Start</li>
</ul>
```




XQuery - New Features

❖ Expressive Power

- Conditional expression
 - if-then-else
- Comparison
 - =, !=, <, <=, >, >=, eq, ne, lt, gt, ge
- Adding elements and attributes to the result
- Selecting and filtering elements
 - for, let, where, order by, return
 - let \$x := (1 to 5) return <test>{\$x}</test>
- Functions and operators
 - Like SQL functions and operators
 - User-defined functions



SQL/XML - New Features

❖ SQL/XML

- An SQL extension for XML
- ISO/ANSI standard: ISO/IEC 9075-14:2003
 - It is different than “SQLXML” of Microsoft SQL Server
- Publishing Functions
 - XMLElements() & XMLAttributes()
 - XMLForest()
 - XMLConcat()
 - XMLAgg()
 - XMLGen()

❖ XML Element, XML Attributes

```
select xmlelement (  
    name "order",  
    xmlattributes (o.oid as "id"),  
    xmlelement (name "signdate", o.contractdate),  
    xmlelement (name "amount", (select sum(orderitem)  
        from orderitems as oi where i.oid = o.oid) )  
    ) from orders as o where status = 'open';
```

```
<order id="4711">  
    <signdate>2002-03-18</signdate>  
    <amount>24000</amount>  
</order>  
<order id="4712">  
    <signdate>2002-03-19</signdate>  
    <amount>44000</amount>  
</order>
```

❖ XMLForest

```
select xmlelement (  
    name "order",  
    xmlforest (o.oid as "id", o.name as "name",  
              o.city as "city"),  
) from orders as o where status = 'open';
```

```
<order>  
  <id>4711</id>  
  <name>steel company</name>  
  <city>Hamburg</city>  
</order>  
<order>  
  <id>4712</id>  
  <name>beer company</name>  
  <city>Munich</city>  
</order>
```



SQL/XML - New Features

❖ XMLConcat

```
select xmlelement (  
    name "order",  
    xmlconcat (  
        xmlelement (name "id", o.oid),  
        xmlelement (name "name", o.name),  
        xmlelement (name "city", o.city),  
    )  
    ) from orders as o where status = 'open';
```

The same result as the previous example



SQL/XML - New Features

❖ XMLAgg

```
select xmlelement (
    name "order", xmlattributes (o.oid as "id"),
    xmlagg (
        xmlelement (name "item", xmlattributes (oi.listnbr as "listnbr"),
            xmlforest (oi.name as "name", oi.quantity as "quantity") )
    order by oi.listnbr ) ) from orders as o where status = 'open';
```

```
<order id="4711">
  <item listnbr="1">
    <name>bike</name>      <quantity>10</quantity>
  </item>
  <item listnbr="2">
    <name>racket</name>   <quantity>5</quantity>
  </item>
</order>
```



SQL/XML - New Features

❖ Datatype

- Datatype is used as same as defined in SQL
- Supported by JDBC 4.0 API
 - The current J2SE 5.0 supports JDBC 3.0

❖ Mapping Rules

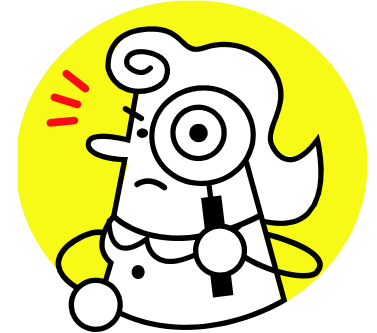
- SQL character \leftrightarrow Unicode
- SQL <identifier> \leftrightarrow XML Name
- SQL Data type \rightarrow XML Schema data type
- SQL table, schema or catalog \rightarrow XML document & Schema



Comparison - New Features

❖ Comparison

- SQL/XML
 - SQL/XML is SQL-centric
 - Useful in relational databases & more familiar
 - SQL has functionality not yet found in XQuery
- XQuery
 - XQuery is XML-centric
 - Useful in native XML environment & powerful
 - No equivalent standard API for XQuery
 - No update or full-text query in XQuery 1.0
- Which type do DBMS vendors or DB users prefer?





Solutions in Market

❖ Solutions in Market

- Two Approaches
- Requirements of DB products



Approaches – Solutions in Market

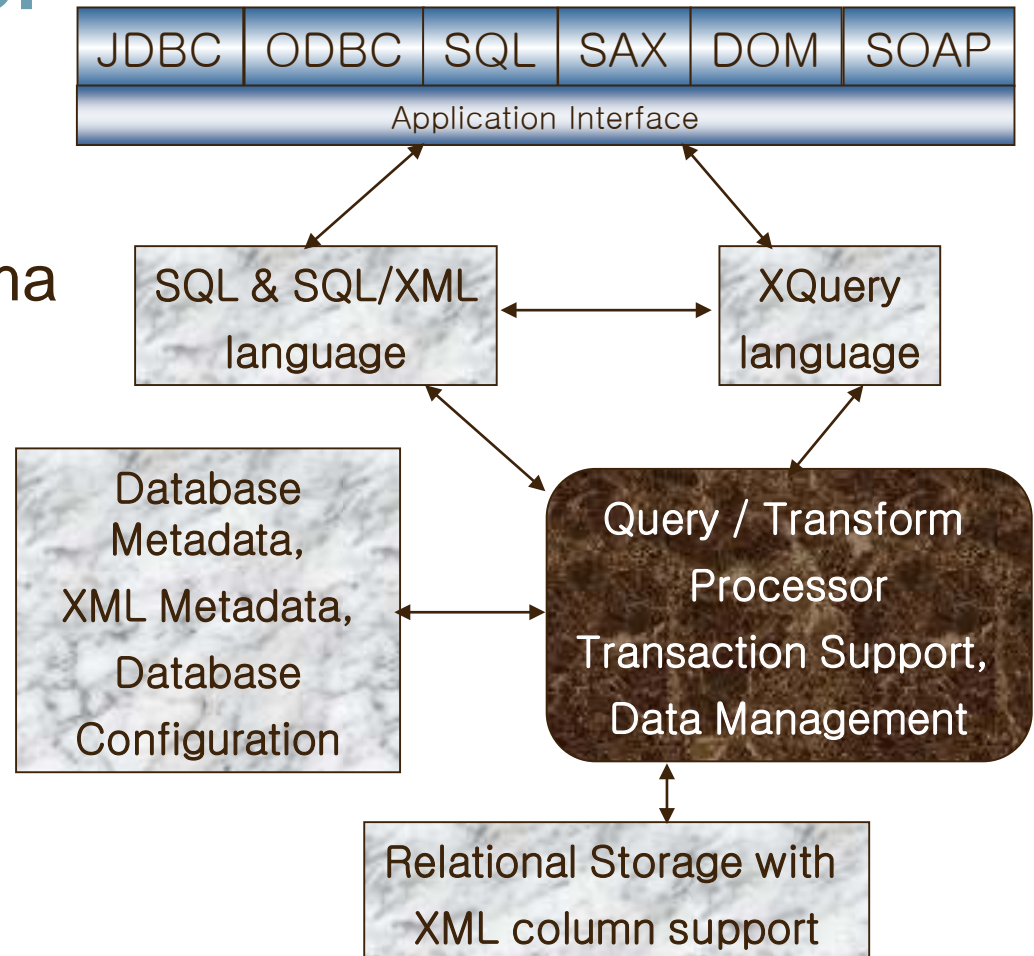
❖ Two Approaches

- Native XML Databases
 - Open source implementations and many commercial vendors
- **XML-enabled Databases**
 - Major commercial vendors
 - **IBM** DB2/Informix/UniData/UniVerse
 - **Microsoft** Access 2002/FoxPro/SQL Server 2000
 - **Oracle** 8i, 9i
 - **Sybase** ASE 12.5
 - and InterSystems Cache, Matisse, Objectivity/DB, Versant enJin, FileMaker, McObject eXtremeDB

Requirements - Solutions in Market

❖ Requirements of DB products

- XML
- DTD or XML Schema
- DOM & XPATH
- XSL





Oracle - Solutions in Market

❖ Oracle's XML products

- XMLDB
- XML-SQL Utility (XSU)
- XML Developer's Kits (XDK)
- XSQL Servlet
- Other products integrated with XML



Oracle - Solutions in Market

❖ Standard Support of Oracle 10g

- W3C **XML Schema** 1.0 Recommendation
- W3C **XPath** 1.0 Recommendation
- W3C **XSL** 1.0 Recommendation
- W3C **DOM** Recommendation Levels 1.0 and 2.0 Core
- Protocol support: HTTP, FTP, IETF WebDAV, as well as Oracle Net
- **Java Servlet** version 2.2, (except that the Servlet WAR file, web.xml is not supported in its entirety, and only one ServletContext and one web-app are currently supported, and stateful servlets are not supported)
- Web Services and Simple Object Access Protocol (**SOAP**). You can access XML stored in the server from SOAP requests
- ISO-ANSI Working Draft for XML-Related Specifications (SQL/XML) [ISO/IEC 9075 Part 14 and ANSI]. Emerging ANSI **SQL/XML** functions to query XML from SQL. The task force defining these specifications falls under the auspices of the International Committee for Information Technology Standards (INCITS). The SQL/XML specification will be fully aligned with SQL:2003
- Java Database Connectivity (**JDBC**)



Conclusion

❖ Conclusion

- Importance of XML
- The useful methods for it

❖ Conclusion

- XML is the exact way to share and exchange information



- SQL/XML & XQuery are critical in integrating non-traditional data sources with existing relational data
- They will eliminate *ad hoc* XML applications and unnecessary effort to integrate data

❖ Reference

- <http://www.w3c.org>
 - <http://www.w3.org/XML>
 - <http://www.w3.org/XML/Schema>
 - <http://www.w3.org/Style/XSL>
 - <http://www.w3.org/XML/Query>
 - <http://www.w3.org/TR/xpath>
- <http://www.sqlx.org>
- <http://www.rpbouret.com>




Thank You !

❖ Empty

- `<xs:element name="product" type="prodtype"/>`
- `<xs:complexType name="prodtype">`
- `<xs:attribute name="prodid"`
- `type="xs:positiveInteger"/>`
- `</xs:complexType>`

- Ex) `<product prodid="34567"/>`

❖ Elements Only

- `<xs:element name="person">`
- `<xs:complexType>`
- `<xs:sequence>`
- `<xs:element name="firstname" type="xs:string"/>`
- `<xs:element name="lastname" type="xs:string"/>`
- `</xs:sequence>`
- `</xs:complexType>`
- `</xs:element>`

- Ex) `<person>`
- `<firstname>John</firstname>`
- `<lastname>Smith</lastname>`
- `</person>`

❖ Text Only

- `<xs:element name="shoesize" type="shoetype"/>`
- `<xs:complexType name="shoetype">`
- `<xs:simpleContent>`
- `<xs:extension base="xs:integer">`
- `<xs:attribute name="country"`
- `type="xs:string" />`
- `</xs:extension>`
- `</xs:simpleContent>`
- `</xs:complexType>`

- Ex) `<shoesize country="france">35</shoesize>`

❖ Mixed

- `<xs:element name="letter" type="lettertype"/>`
- `<xs:complexType name="lettertype" mixed="true">`
- `<xs:sequence>`
- `<xs:element name="name" type="xs:string"/>`
- `<xs:element name="orderid"`
`type="xs:positiveInteger"/>`
- `<xs:element name="shipdate" type="xs:date"/>`
- `</xs:sequence>`
- `</xs:complexType>`

- Ex) `<letter> Dear Mr.<name>John Smith</name>.`
`Your order <orderid>1032</orderid> will be shipped`
`on <shipdate>2001-07-13</shipdate>. </letter>`

❖ Indicator - all

- `<xs:element name="person">`
- `<xs:complexType>`
- `<xs:all>`
- `<xs:element name="firstname" type="xs:string"/>`
- `<xs:element name="lastname" type="xs:string"/>`
- `</xs:all>`
- `</xs:complexType>`
- `</xs:element>`

❖ Indicator – choice

- `<xs:element name="person">`
- `<xs:complexType>`
- `<xs:choice>`
- `<xs:element name="employee" type="employee"/>`
- `<xs:element name="member" type="member"/>`
- `</xs:choice>`
- `</xs:complexType>`
- `</xs:element>`

❖ Indicator - sequence

- `<xs:element name="person">`
- `<xs:complexType>`
- `<xs:sequence>`
- `<xs:element name="firstname" type="xs:string"/>`
- `<xs:element name="lastname" type="xs:string"/>`
- `</xs:sequence>`
- `</xs:complexType>`
- `</xs:element>`

❖ Indicator - occurrence

- `<xs:element name="person">`
- `<xs:complexType>`
- `<xs:sequence>`
- `<xs:element name="full_name" type="xs:string"/>`
- `<xs:element name="child_name" type="xs:string"`
- `maxOccurs="10"/>`
- `</xs:sequence>`
- `</xs:complexType>`
- `</xs:element>`



Appendix

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="stringtype">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
  <xs:simpleType name="inttype">
    <xs:restriction base="xs:positiveInteger"/>
  </xs:simpleType>
  <xs:simpleType name="dectype">
    <xs:restriction base="xs:decimal"/>
  </xs:simpleType>
  <xs:simpleType name="orderidtype">
    <xs:restriction base="xs:string">
      <xs:pattern value="[0-9]{6}"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:complexType name="shiptotype">
    <xs:sequence>
      <xs:element name="name" type="stringtype"/>
      <xs:element name="address" type="stringtype"/>
      <xs:element name="city" type="stringtype"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Appendix

```
<xs:element name="country" type="stringtype"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="itemtype">
  <xs:sequence>
    <xs:element name="title" type="stringtype"/>
    <xs:element name="note" type="stringtype" minOccurs="0"/>
    <xs:element name="quantity" type="inttype"/>
    <xs:element name="price" type="dectype"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="shipordertype">
  <xs:sequence>
    <xs:element name="orderperson" type="stringtype"/>
    <xs:element name="shipto" type="shiptotype"/>
    <xs:element name="item"
      minOccurs="1" maxOccurs="unbounded" type="itemtype"/>
  </xs:sequence>
  <xs:attribute name="orderid" type="orderidtype" use="required"/>
</xs:complexType>
<xs:element name="shiporder" type="shipordertype"/>
</xs:schema>
```