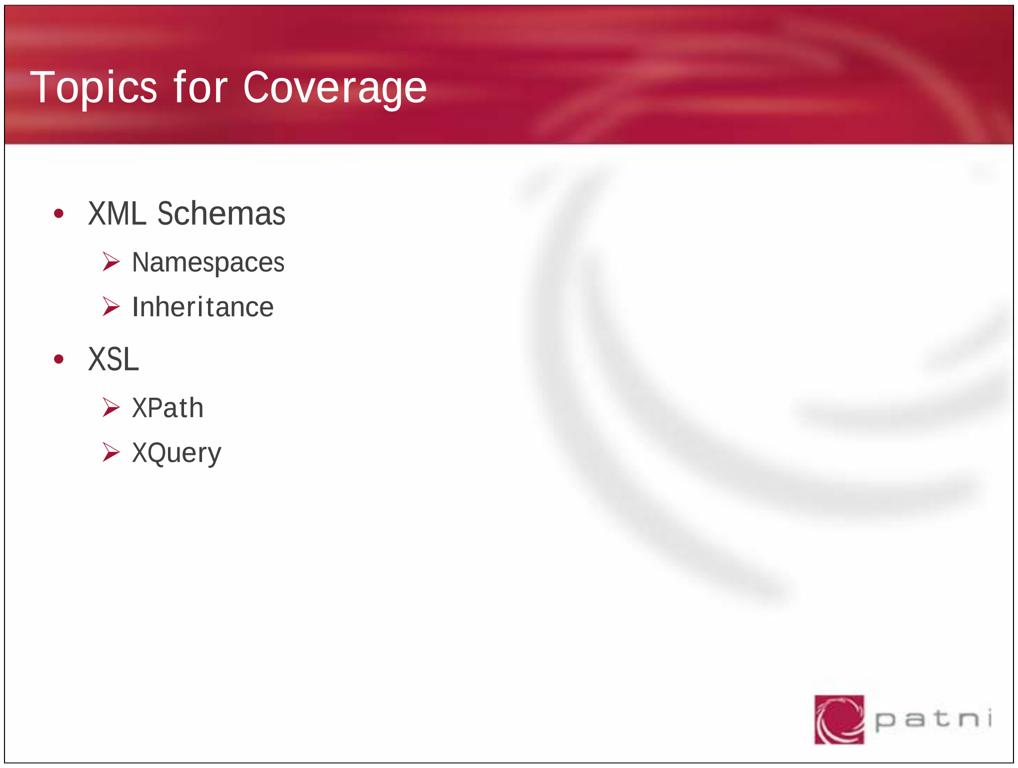
Advanced XML

PACE

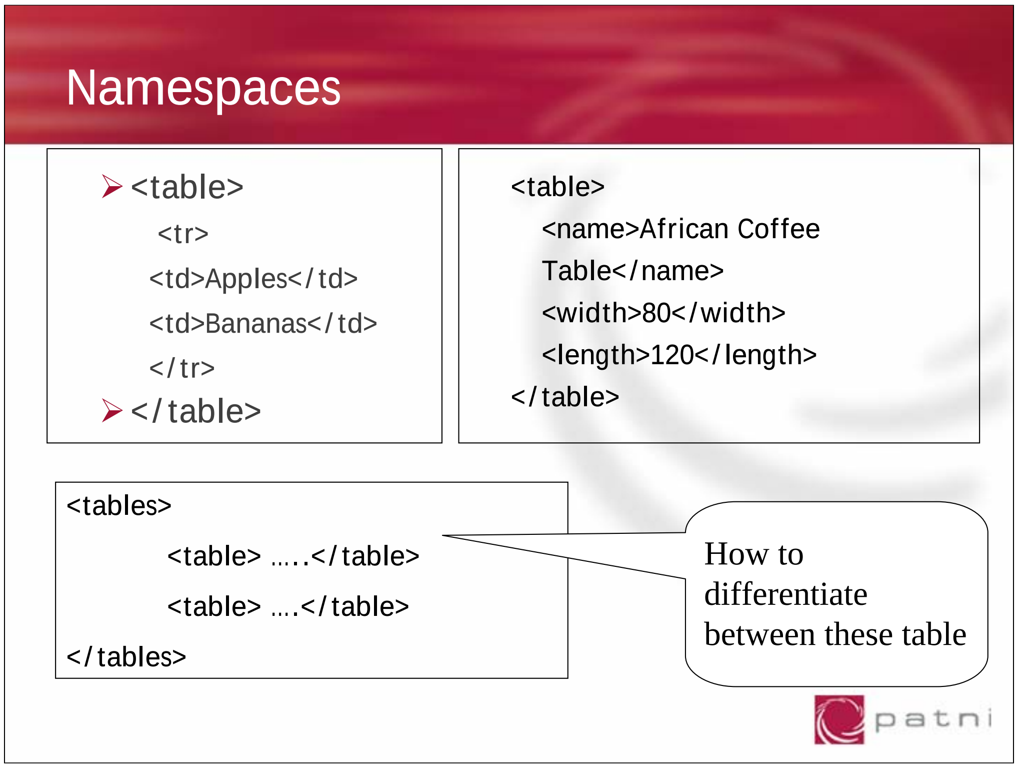
Patni Academy for Competency Enhancement



Topics for Coverage

- XML Schemas
  - Namespaces
  - Inheritance
- XSL
  - XPath
  - XQuery

patni



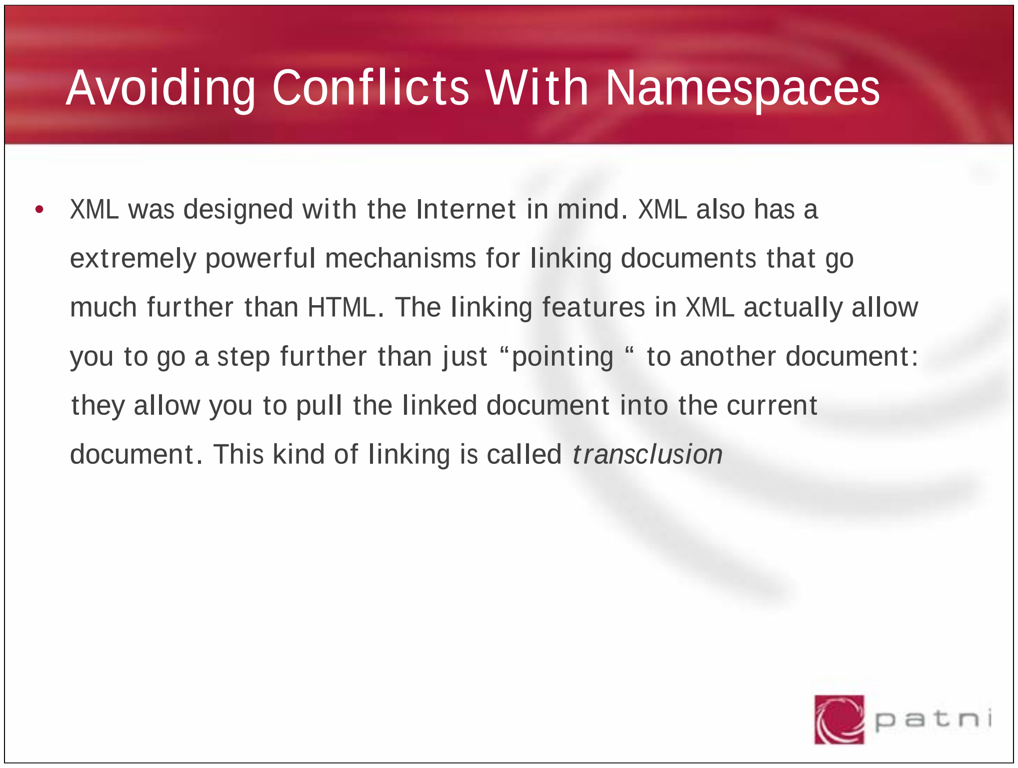
Namespaces

```
➢ <table>
     <tr>
    <td>Apples</td>
    <td>Bananas</td>
    </tr>
➢ </table>
```

```
<table>
   <name>African Coffee
   Table</name>
   <width>80</width>
   <length>120</length>
</table>
```

```
<tables>
          <table> …..</table>
          <table> ….</table>
</tables>
```

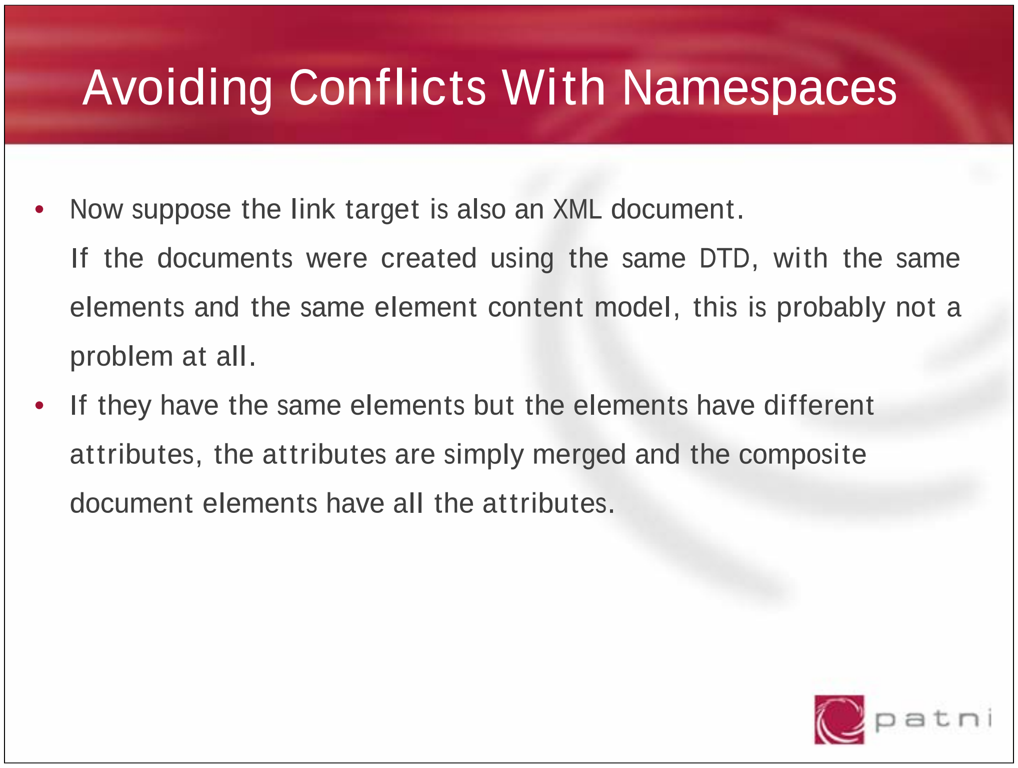
How to differentiate between these table



Avoiding Conflicts With Namespaces

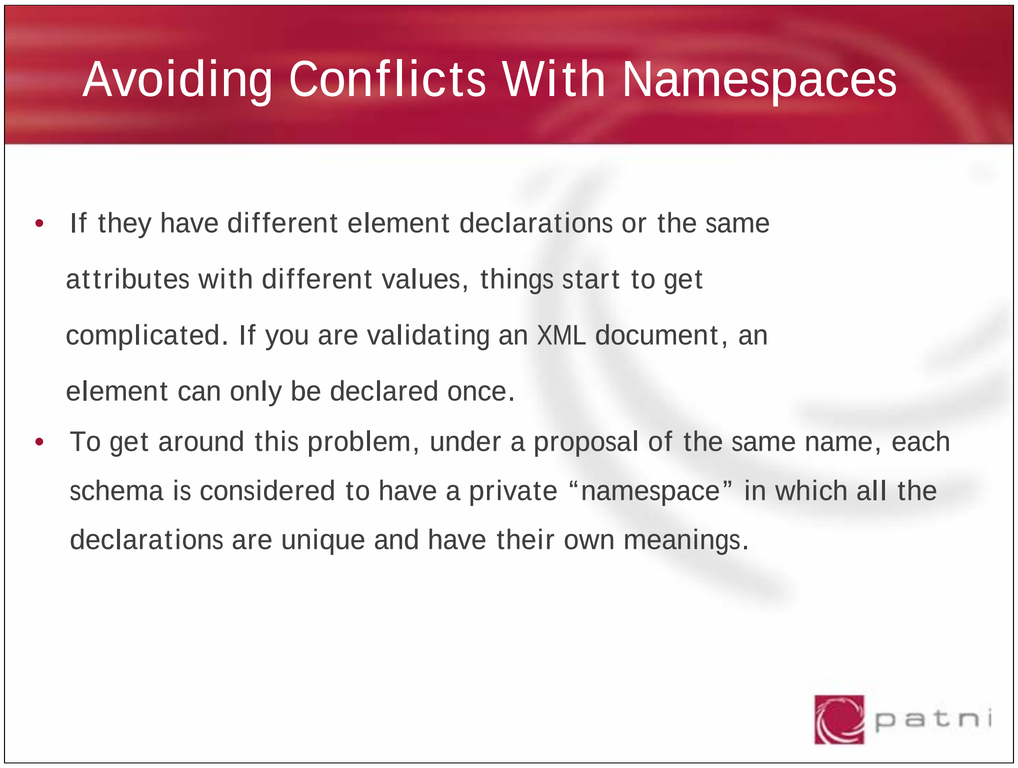
- XML was designed with the Internet in mind. XML also has a extremely powerful mechanisms for linking documents that go much further than HTML. The linking features in XML actually allow you to go a step further than just "pointing " to another document: they allow you to pull the linked document into the current document. This kind of linking is called *transclusion*

patni



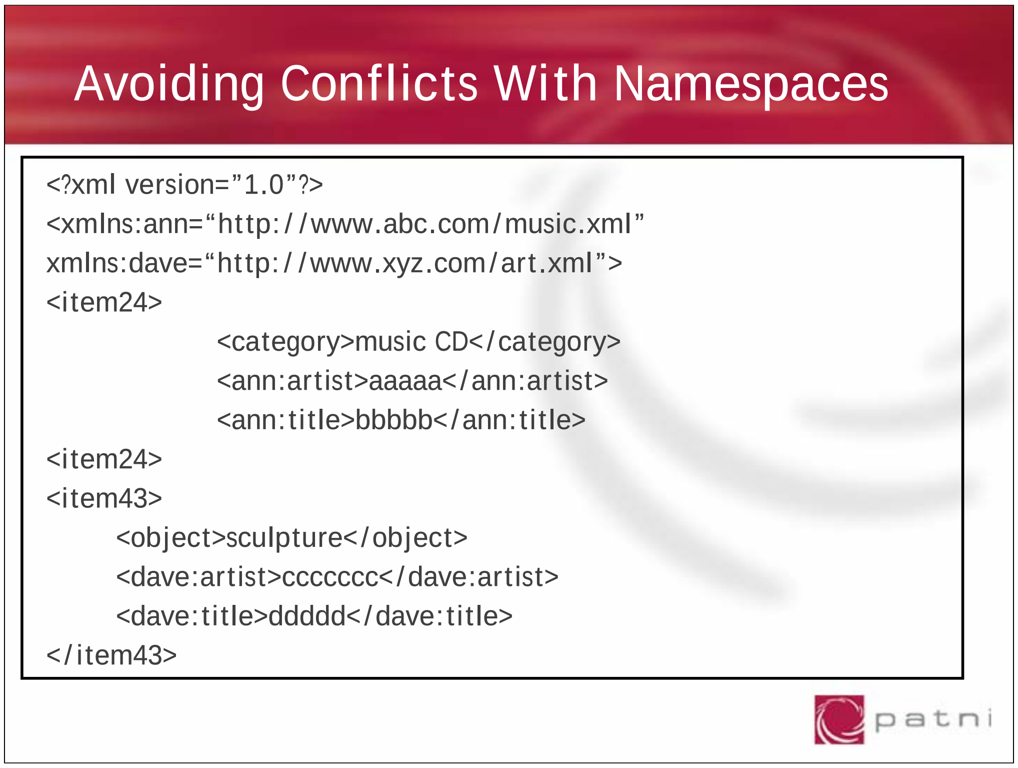
Avoiding Conflicts With Namespaces

- Now suppose the link target is also an XML document.
  If the documents were created using the same DTD, with the same elements and the same element content model, this is probably not a problem at all.
- If they have the same elements but the elements have different attributes, the attributes are simply merged and the composite document elements have all the attributes.



Avoiding Conflicts With Namespaces

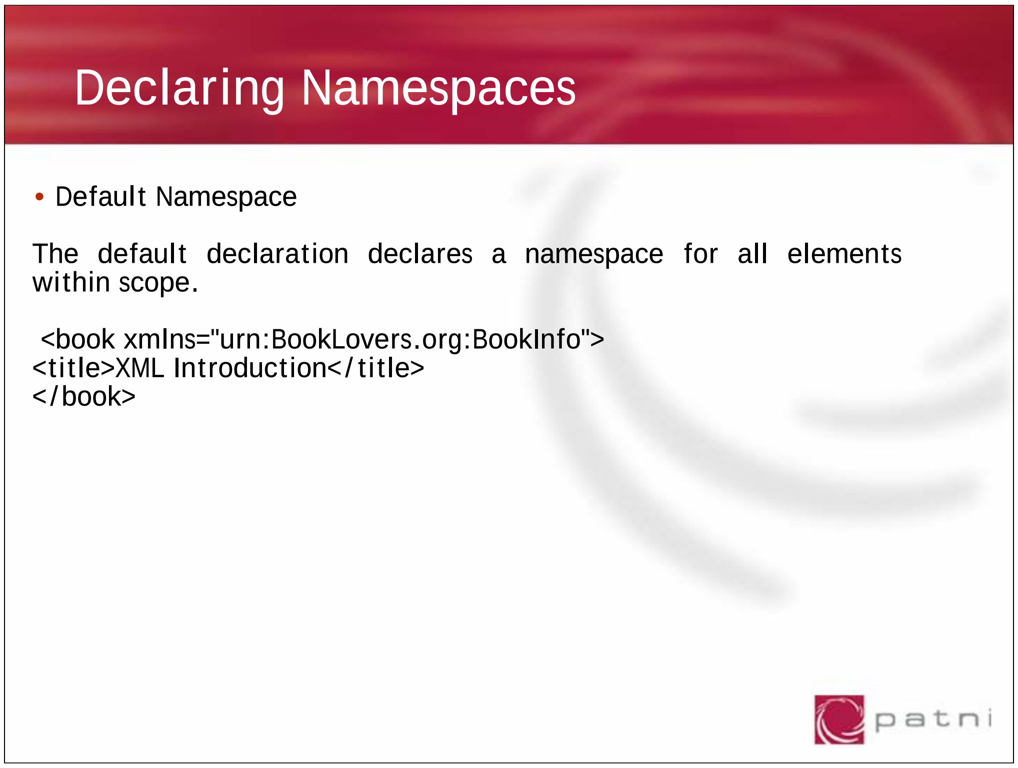
- If they have different element declarations or the same attributes with different values, things start to get complicated. If you are validating an XML document, an element can only be declared once.
- To get around this problem, under a proposal of the same name, each schema is considered to have a private "namespace" in which all the declarations are unique and have their own meanings.



Avoiding Conflicts With Namespaces

```
<?xml version=”1.0”?>
<xmlns:ann=“http://www.abc.com/music.xml”
xmlns:dave=“http://www.xyz.com/art.xml”>
<item24>
            <category>music CD</category>
            <ann:artist>aaaaa</ann:artist>
            <ann:title>bbbbb</ann:title>
<item24>
<item43>
     <object>sculpture</object>
     <dave:artist>ccccccc</dave:artist>
     <dave:title>ddddd</dave:title>
</item43>
```

patni



Declaring Namespaces

- Default Namespace

The default declaration declares a namespace for all elements within scope.

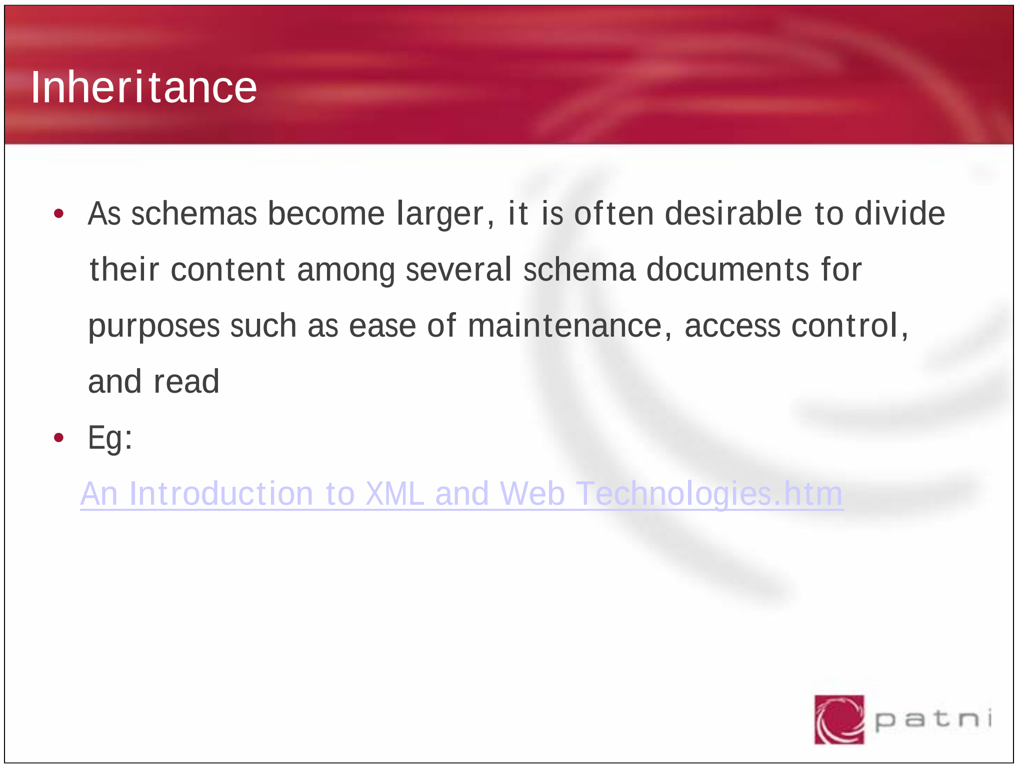
```
 <book xmlns="urn:BookLovers.org:BookInfo">
<title>XML Introduction</title>
</book>
```

Declaring Namespaces

- Explicit Declaration
  The following example declares "bk" and "money" to be shorthand for the full names of their respective namespaces. All elements beginning with "bk:" or "money:" are considered to be from the namespace "urn:BookLovers.org:BookInfo" or “urn:Finance:Money," respectively.

```
<bk:book xmlns:bk="urn:BookLovers.org:BookInfo"
            xmlns:money="urn:Finance:Money">
   <bk:title>A Suitable Boy</bk:title>
   <money:PRICE money:currency="US"Dollar">22.95</money:PRICE>
</bk:BOOK>
```

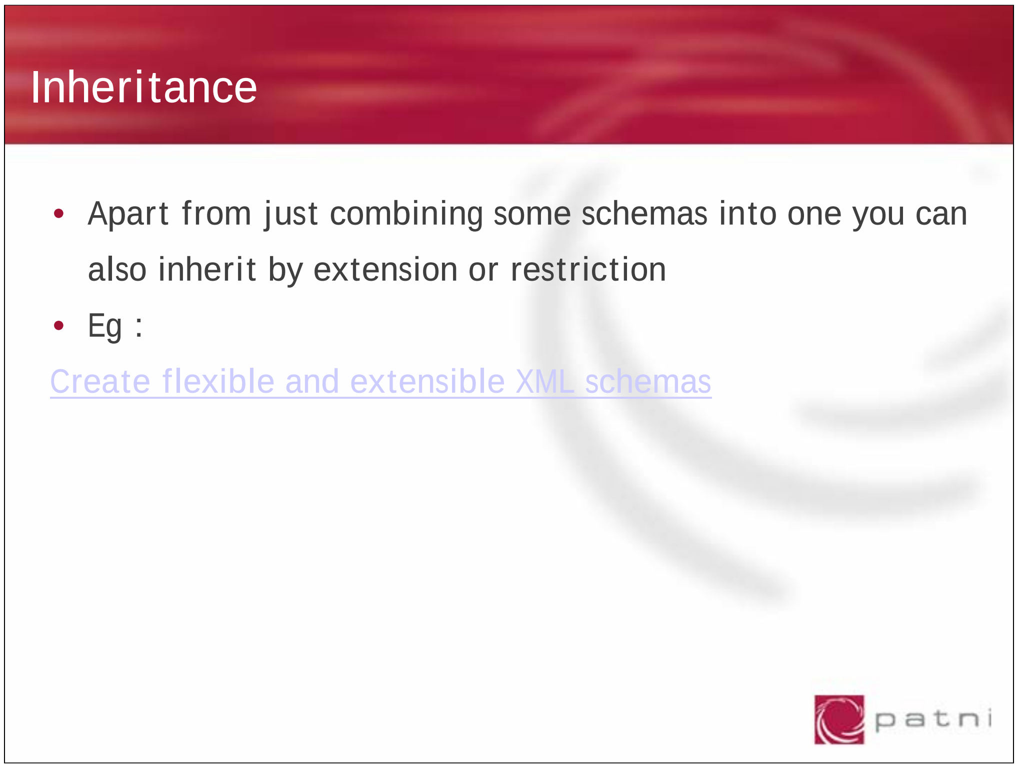
patni



Inheritance

- As schemas become larger, it is often desirable to divide their content among several schema documents for purposes such as ease of maintenance, access control, and read
- Eg:

  An Introduction to XML and Web Technologies.htm



Inheritance

- Apart from just combining some schemas into one you can also inherit by extension or restriction
- Eg :

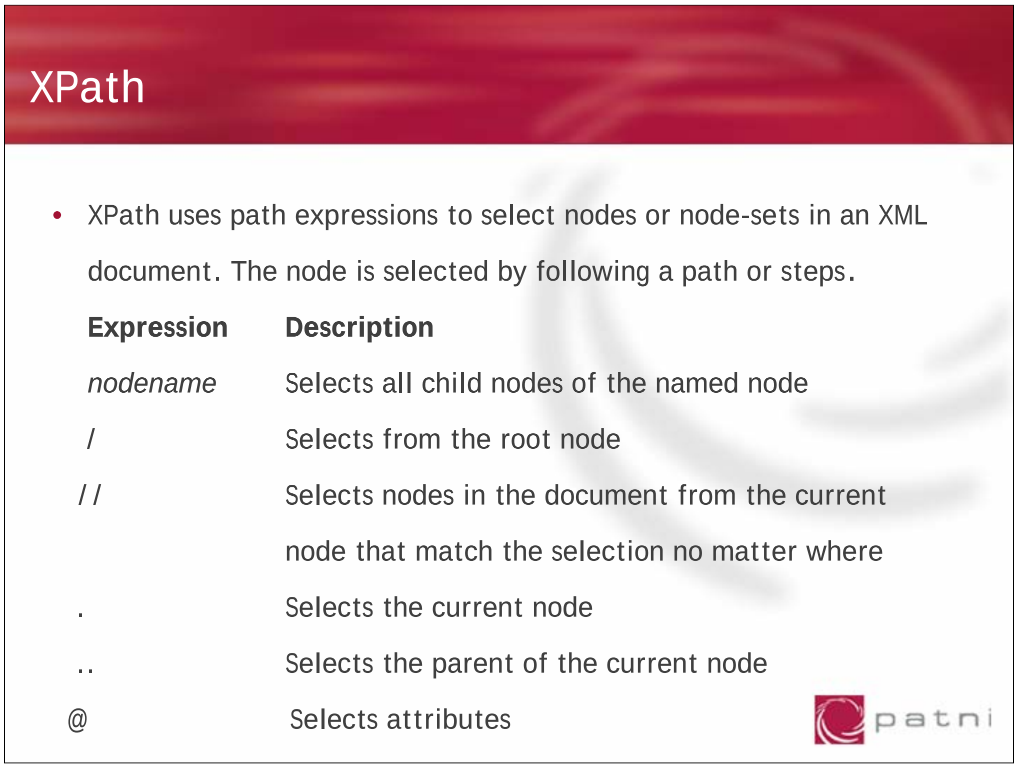
Create flexible and extensible XML schemas

patni

XPath

- XPath is a language for finding information in an XML document. XPath is used to navigate through elements and attributes in an XML document
- XPath uses path expressions to select nodes or node-sets in an XML document
- In XPath, there are seven kinds of nodes: element, attribute, text, namespace, processing-instruction, comment, and document (root) nodes

patni

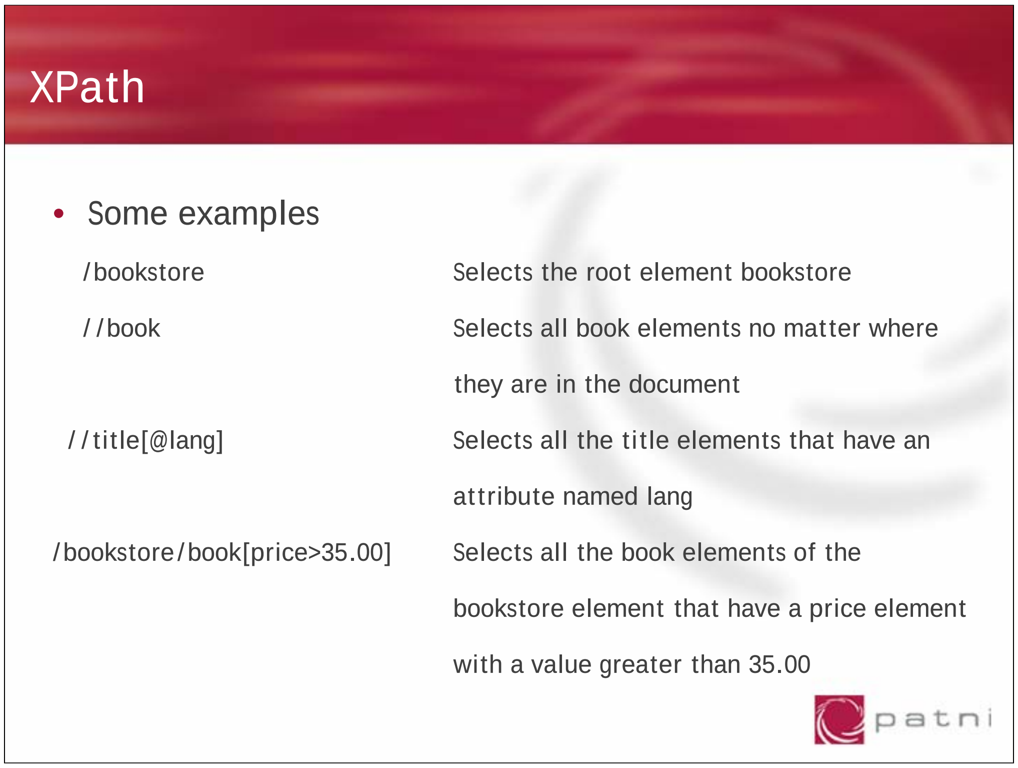


XPath

- XPath uses path expressions to select nodes or node-sets in an XML document. The node is selected by following a path or steps.

Expression	**Description**
nodename	Selects all child nodes of the named node
/	Selects from the root node
//	Selects nodes in the document from the current node that match the selection no matter where
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes

patni

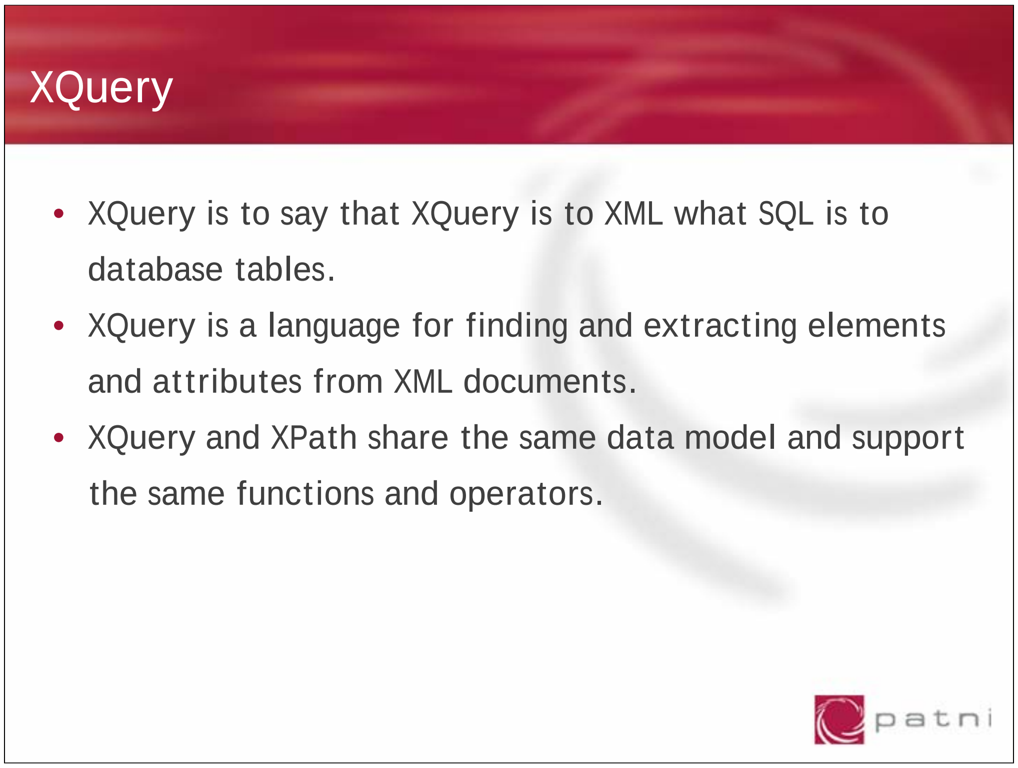


XPath

- Some examples

/bookstore	Selects the root element bookstore
//book	Selects all book elements no matter where they are in the document
//title[@lang]	Selects all the title elements that have an attribute named lang
/bookstore/book[price>35.00]	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00

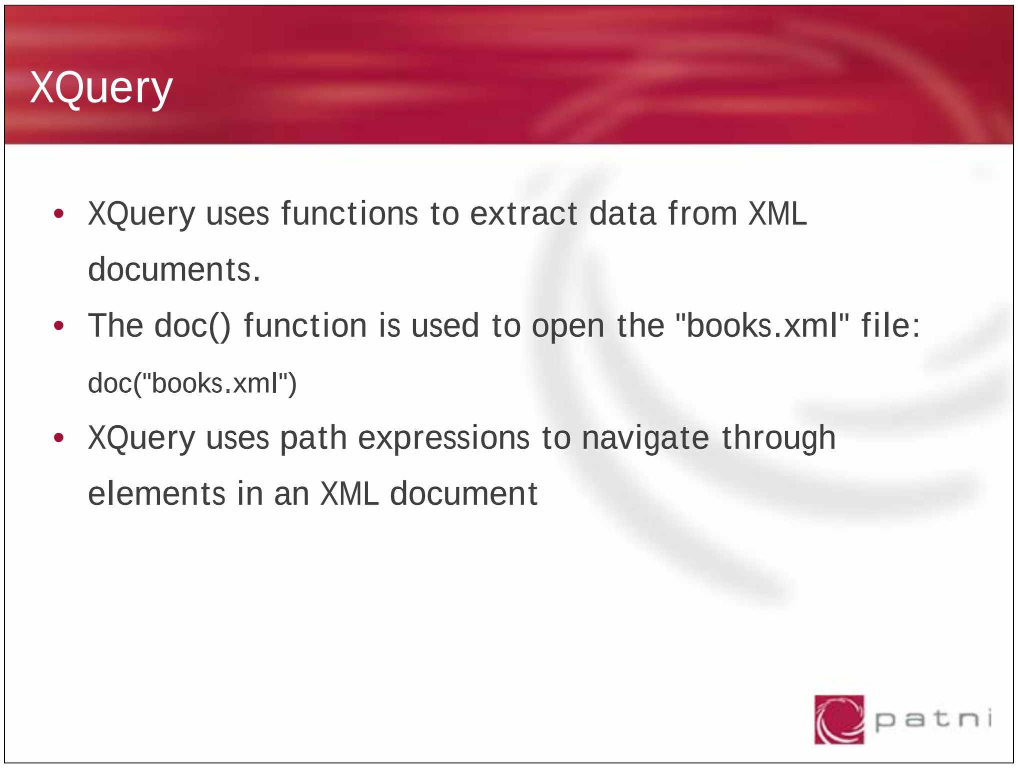
patni



XQuery

- XQuery is to say that XQuery is to XML what SQL is to database tables.
- XQuery is a language for finding and extracting elements and attributes from XML documents.
- XQuery and XPath share the same data model and support the same functions and operators.

patni

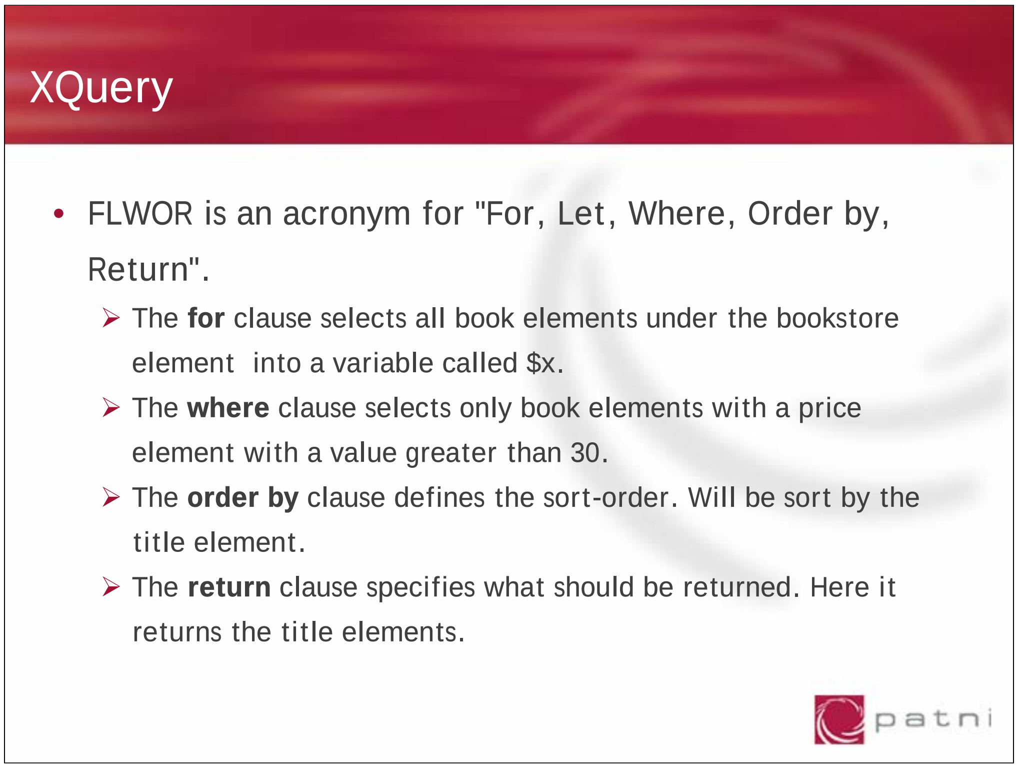


XQuery

- XQuery uses functions to extract data from XML documents.
- The doc() function is used to open the "books.xml" file:

  doc("books.xml")
- XQuery uses path expressions to navigate through elements in an XML document

patni



XQuery

- FLWOR is an acronym for "For, Let, Where, Order by, Return".
  - The **for** clause selects all book elements under the bookstore element into a variable called $x.
  - The **where** clause selects only book elements with a price element with a value greater than 30.
  - The **order by** clause defines the sort-order. Will be sort by the title element.
  - The **return** clause specifies what should be returned. Here it returns the title elements.

Thanks

PACE

Patni Academy for Competency Enhancement