Sabrina already created a document containing a list of degrees currently offered by the Information Technology department. She wants the document validated based on a schema you create.

Complete the following:

1. Using your XML editor, open the **degreestxt.xml** and **degreestxt.xsd** files from the xml03 ▸ review folder, enter ***your name*** and ***today's date*** in the comment section of each file, and then save the files as **degrees.xml** and **degrees.xsd**, respectively.
2. In the degrees.xsd file, add the root schema element to the document and declare the XML Schema namespace using the `xs` prefix, and then save your work.
3. In the degrees.xml file, attach the schema file **degrees.xsd** to this instance document, indicating that the schema and instance document do not belong to any namespace, and then save your work.
4. In the degrees.xsd file, create the following named simple types:
   a. **`idType`**, based on the `ID` data type and restricted to the regular expression pattern `IT\d{2}-\d{3}-\d{3}`
   b. **`codeType`**, based on the `string` data type and restricted to the following values—`MP`, `SP`, `WPA`
5. Declare the **`degrees`** element containing the child element **`degree`**.
6. Declare the **`degree`** element containing the following sequence of nested child elements—`title`, `approvalDate`, `effectiveDate`, `summary`, `coordinator`, and `comment`. Set the following properties for the nested elements:
   a. All of the child elements should contain string data except the `approvalDate` and `effectiveDate` elements, which contain dates. The `degree` element should also support two required attributes—`degreeID` and `degreeCode`. The `degreeID` attribute contains `idType` data, while the `degreeCode` attribute contains `codeType` data.
   b. The `degree` element must occur at least once, but its upper limit is unbounded. The `approvalDate` element is optional. The `comment` element is optional, and it may occur multiple times. All other elements are assumed to occur only once.
   c. Each `comment` element requires a `date` attribute of the `date` data type.
7. Save your changes to the degrees.xsd file, and then validate the schema document. Correct any errors you find.
8. Validate the degrees.xml file against the schema document you created. Correct any validation errors you discover in the instance document.

APPLY

Case Problem 1

Data Files needed for this Case Problem: catalogtxt.xml, catalogtxt.xsd

The Our Lady of Bergen Historical Society Sharon Strattan is an archivist at the Our Lady of Bergen Historical Society in Bergenfield, New Jersey. The historical society is exploring how to transfer its listings to XML format, and Sharon has begun by creating a sample document of the society's extensive collection of photos. As part of this process, she's asked for your help in developing the schema that will be used to validate the XML documents. She has created a sample document to work on. Eventually, your work will be used in a much larger system. The structure of the sample document is shown in Figure 3-40.

Figure 3-40 The catalog vocabulary structure

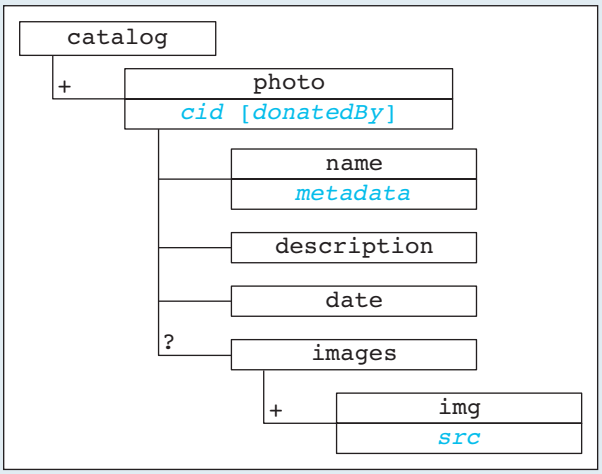

Figure 3-41 describes the elements and attributes in this sample document as well as the rules that govern the data that can be entered into a valid document.

Figure 3-41 The catalog vocabulary

Element or Attribute	Description
`catalog`	The root element
`photo`	The collection of information about a photo
`cid`	The ID number of the catalog with the format c####, where # is a digit
`donatedBy`	The name of the donor
`name`	The name of the photo
`metadata`	The metadata for the photo
`description`	The description of the photo
`date`	The approximate date of the photo
`images`	The collection of img elements
`img`	The element that references the image file
`src`	The source file containing the image; must end with .jpg

Your job will be to express this document structure and set of rules in terms of the XML Schema language, and then to validate Sharon's document based on the schema you create.

Complete the following:

1. Using your XML editor, open the **catalogtxt.xml** and **catalogtxt.xsd** files from the xml03 ▸ case1 folder, enter ***your name*** and ***today's date*** in the comment section of each file, and then save the files as **catalog.xml** and **catalog.xsd**, respectively.
2. Go to the catalog.xsd file in your text editor. Add the root schema element to the document and declare the XML Schema namespace using the `xs` prefix.
3. Attach the schema file **catalog.xsd** to the instance document, indicating that the schema and instance document do not belong to any namespace.

4. Create the following named simple types:
   a. **cidType**, based on the ID data type and restricted to the regular expression pattern c\d{4}
   b. **srcType**, based on the string data type and restricted to the regular expression pattern [a-zA-Z0-9]+.jpg
5. Declare the **catalog** element containing the child element photo. The photo element must occur at least once, but its upper limit is unbounded.
6. Declare the **photo** element containing the following sequence of nested child elements—name, description, date, and images. Set the following properties for the nested elements:
   a. All of the child elements should contain string data. The name element should also support the metadata attribute.
   b. The cid attribute is required. The donatedBy attribute is optional.
7. Declare the **img** element. It has no content and contains a required attribute, src.
8. Declare the following attributes and elements:
   a. The attribute metadata must have the string data type.
   b. The attribute cid must have the cidType data type.
   c. The attribute src must have the srcType data type.
   d. The attribute donatedBy must have the string data type.
   e. The element description must have the string data type.
   f. The element date must have the string data type.
9. Save your changes to the catalog.xsd file, and then validate the schema. Continue to correct any validation errors you discover until the schema validates.
10. Validate the catalog.xml file against the schema. Continue to correct any validation errors you discover until the instance document validates.

APPLY

Case Problem 2

Data Files needed for this Case Problem: mdpbatxt.xml, mdpbatxt.xsd

Midwest Developmental Pipe Band Association Jacob St. John works as a coordinator for the Midwest Developmental Pipe Band Association (MDPBA) and is responsible for coordinating competitions for the MDPBA's many developmental pipe bands in the Midwest. Part of Jacob's job is to maintain a document that lists competition entries for each pipe band. As part of this process, he's asked for your help with developing the schema that will be used to validate the XML documents. He has created a sample document to work on. The structure of the sample document is shown in Figure 3-42.