CertsCare

Free Questions for I10-003 by certscare

Shared by Hammond on 06-06-2022

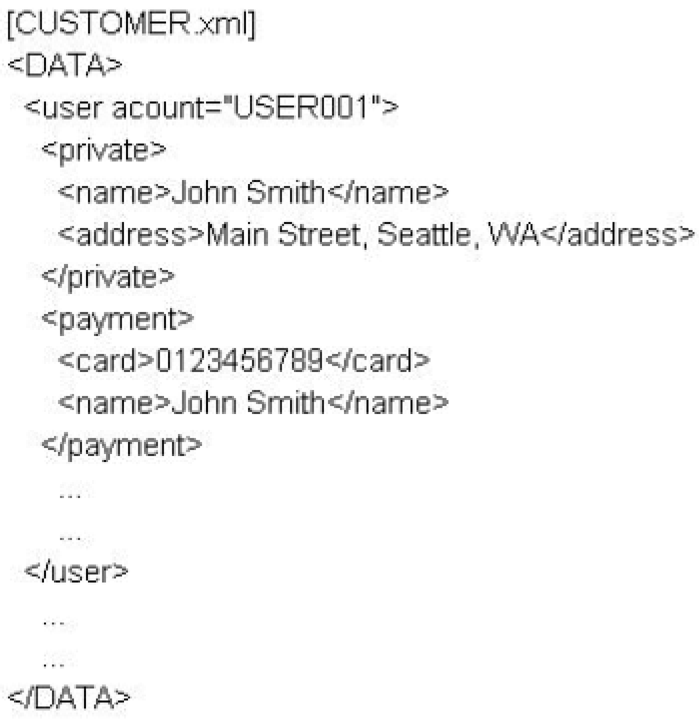
For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

A certain store engages in Internet commerce, managing customer information via XMLDB. Customers register as a user through a webpage, and are allowed to view their own information so they can edit their information themselves through a webpage interface. The store's Web application saves the customer information in an XMLDB, and retrieves data from the XMLDB as necessary. The XML data including customer information is as shown in [CUSTOMER.xml] referenced in a separate window.



```
[CUSTOMER.xml]
<DATA>
  <user acount="USER001">
    <private>
      <name>John Smith</name>
      <address>Main Street, Seattle, WA</address>
    </private>
    <payment>
      <card>0123456789</card>
      <name>John Smith</name>
    </payment>
      ...
      ...
  </user>
    ...
    ...
</DATA>
```

The XMLDB account when the Web application connects to the XMLDB is WEBAPP.

A person at the store is in charge of processing payments (access to all registered customer information), and this person's XMLDB account is COUNTER.

A person at the store is in charge of product shipments (access to all registered customer information except for payment information ("payment element")), and this person's XMLDB account is SHIPPER.

Do not consider XMLDB accounts other than those noted above.

Each account authorization in the XMLDB is presently configured as follows: The WEBAPP account has permission to update and view [CUSTOMER xml]

Other accounts have permission to view [CUSTOMER.xml]

Which is the most appropriate method in this situation regarding XMLDB account authorizations'?

Assume that this XMLDB has a view creation function (function to show only certain XML data in response to a certain query)

Options:

A- When saving data into the XMLDB, all user element content should be encrypted, and all XMLDB accounts should be given permission for decryption

B- When saving data into the XMLDB, all payment element content should be encrypted, and only the COUNTER account should be given permission for decryption

C- You should create a view (PAYMENT_VIEW) to show only payment element information, providing the COUNTER account with permission to view PAYMENT_VIEW

D- You should create a view (SHIP_V1EW) to show information other than payment element information, providing the SHIPPER account with permission to view SHIP_V1EW, and prohibiting the SHIPPER account from viewing [CUSTOMER.xml]

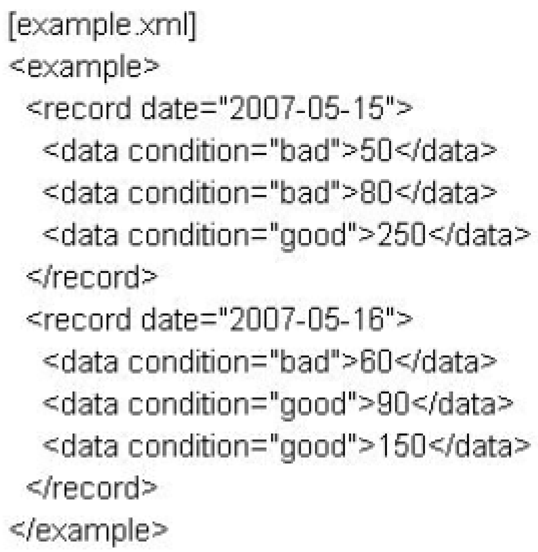
Answer:

D

Question 2

Question Type: MultipleChoice

Select the correct result of executing the following [XQuery] on [example xml] referenced in a separate window.



```
[example.xml]
<example>
  <record date="2007-05-15">
    <data condition="bad">50</data>
    <data condition="bad">80</data>
    <data condition="good">250</data>
  </record>
  <record date="2007-05-16">
    <data condition="bad">60</data>
    <data condition="good">90</data>
    <data condition="good">150</data>
  </record>
</example>
```

[XQuery]

```
{

for $record in fn:doc("example.xml")/example/record

return

{ fn:data($record/data) }

}
```

Options:

A- <result>

```
<record>50</record>
<record>60</record>
</result>
```

B- <result>

```
<record>5080250</record>
<record>6090150</record>
</result>
```

C- <result>

```
<record>50 80 250</record>
<record>60 90 150</record>
</result>
```

D- An error occurs

Answer:

D

Question 3

Question Type: MultipleChoice

Select the correct result of executing the [XOuery] on [example xml] referenced in a separate window.

[example.xml]

```
<example>
  <record>
    <dept>Sales Department</dept>
    <group>Group No1</group>
    <title>Group Leader</title>
    <name>John Smith</name>
  </record>
  <record>
    <dept>Engineering Department</dept>
    <name>Harold Jones</name>
  </record>
</example>
```

[XQuery]

```
declare function local:func($n) {
  if (fn:name($n) = "example") then
    for $c in $n/* return local:func($c)
  else if (fn:name($n) = "record") then
    element { fn:name($n) }
            { for $c in $n/* return local:func($c) }
  else if (fn:name($n) = "name") then $n
  else ( )
};
<result>{
  local:func(fn:doc("example.xml"))
}</result>
```

Options:

A-
```
<result/>
```

B-
```
<result>
<record/>
<record/>
</result>
```

C-
```
<result>
<record>
<name/>
</record>
<record>
<name/>
</record>
</result>
```

D-
```
<result>
<record>
<name>John Smith</name>
</record>
<record>
<name>Harold Jones</name>
</record>
</result>
```

Answer:

A

Question 4

Question Type: MultipleChoice

See separate window.

[PRODUCTS.xml]

```
<PRODUCTS>
  <record>
    <PID>C001</PID>
    <PRODUCT>Chair</PRODUCT>
    <PRICE>6000</PRICE>
  </record>
  <record>
    <PID>T001</PID>
    <PRODUCT>Table</PRODUCT>
    <PRICE>20000</PRICE>
  </record>
</PRODUCTS>
```

[STOCKLIST.xml]

```
<STOCKLIST>
  <record>
    <STOCKROOM>Yokohama</STOCKROOM>
    <PID>C001</PID>
    <QUANTITY>10</QUANTITY>
  </record>
  <record>
    <STOCKROOM>Yokohama</STOCKROOM>
    <PID>T001</PID>
    <QUANTITY>3</QUANTITY>
  </record>
  <record>
    <STOCKROOM>Kawasaki</STOCKROOM>
    <PID>T001</PID>
    <QUANTITY>1</QUANTITY>
  </record>
</STOCKLIST>
```

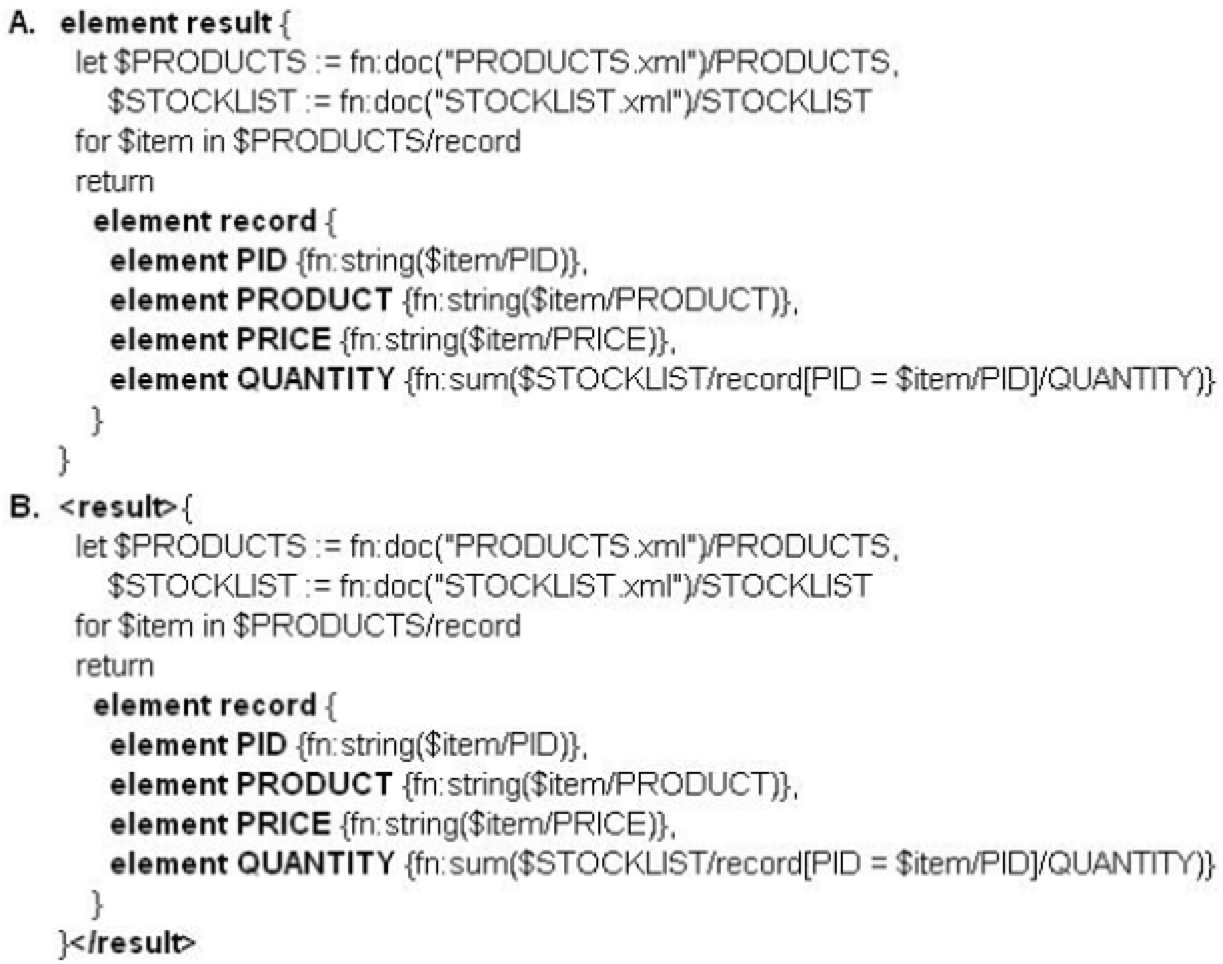
[Output Result]

```
<result>
  <record>
    <PID>C001</PID>
    <PRODUCT>Chair</PRODUCT>
    <PRICE>6000</PRICE>
    <QUANTITY>10</QUANTITY>
  </record>
  <record>
    <PID>T001</PID>
```

[PRODUCTS xml] (separate window) and [STOCKLIST.xml] (separate window) are output in XML format from RDB (relational database) data.

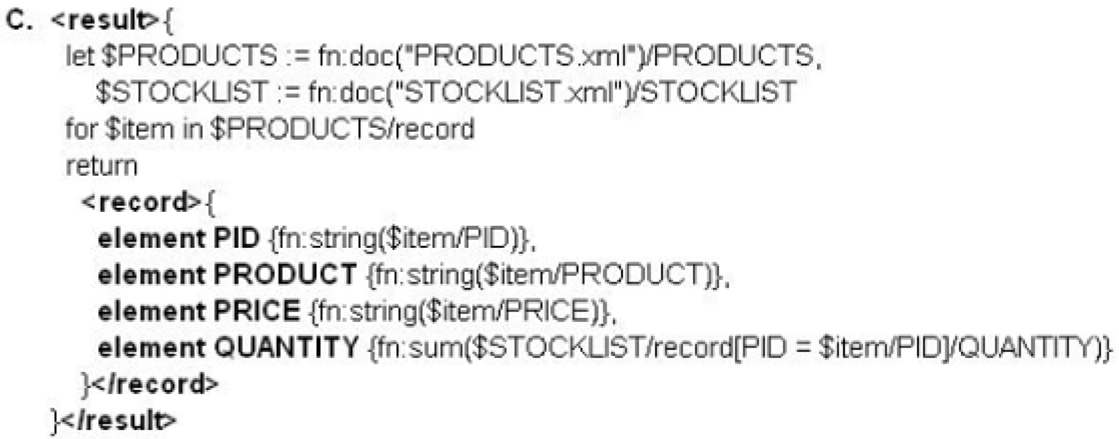
Assume that you wish to use an XOuery processor to get [Output Result] (separate window) from this XML data.

Which of the following is an XOuery that cannot retrieve [Output Result]?

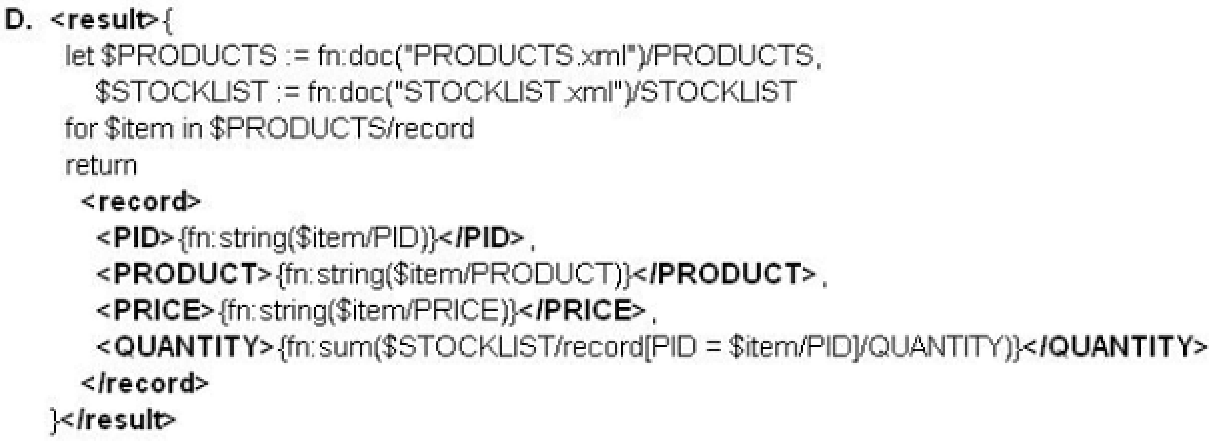


A.
```
element result {
  let $PRODUCTS := fn:doc("PRODUCTS.xml")/PRODUCTS,
      $STOCKLIST := fn:doc("STOCKLIST.xml")/STOCKLIST
  for $item in $PRODUCTS/record
  return
    element record {
      element PID {fn:string($item/PID)},
      element PRODUCT {fn:string($item/PRODUCT)},
      element PRICE {fn:string($item/PRICE)},
      element QUANTITY {fn:sum($STOCKLIST/record[PID = $item/PID]/QUANTITY)}
    }
}
```

B.
```
<result>{
  let $PRODUCTS := fn:doc("PRODUCTS.xml")/PRODUCTS,
      $STOCKLIST := fn:doc("STOCKLIST.xml")/STOCKLIST
  for $item in $PRODUCTS/record
  return
    element record {
      element PID {fn:string($item/PID)},
      element PRODUCT {fn:string($item/PRODUCT)},
      element PRICE {fn:string($item/PRICE)},
      element QUANTITY {fn:sum($STOCKLIST/record[PID = $item/PID]/QUANTITY)}
    }
}</result>
```



```
C. <result>{
     let $PRODUCTS := fn:doc("PRODUCTS.xml")/PRODUCTS,
         $STOCKLIST := fn:doc("STOCKLIST.xml")/STOCKLIST
     for $item in $PRODUCTS/record
     return
       <record>{
         element PID {fn:string($item/PID)},
         element PRODUCT {fn:string($item/PRODUCT)},
         element PRICE {fn:string($item/PRICE)},
         element QUANTITY {fn:sum($STOCKLIST/record[PID = $item/PID]/QUANTITY)}
       }</record>
   }</result>
```



```
D. <result>{
     let $PRODUCTS := fn:doc("PRODUCTS.xml")/PRODUCTS,
         $STOCKLIST := fn:doc("STOCKLIST.xml")/STOCKLIST
     for $item in $PRODUCTS/record
     return
       <record>
         <PID>{fn:string($item/PID)}</PID> ,
         <PRODUCT>{fn:string($item/PRODUCT)}</PRODUCT> ,
         <PRICE>{fn:string($item/PRICE)}</PRICE> ,
         <QUANTITY>{fn:sum($STOCKLIST/record[PID = $item/PID]/QUANTITY)}</QUANTITY>
       </record>
   }</result>
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

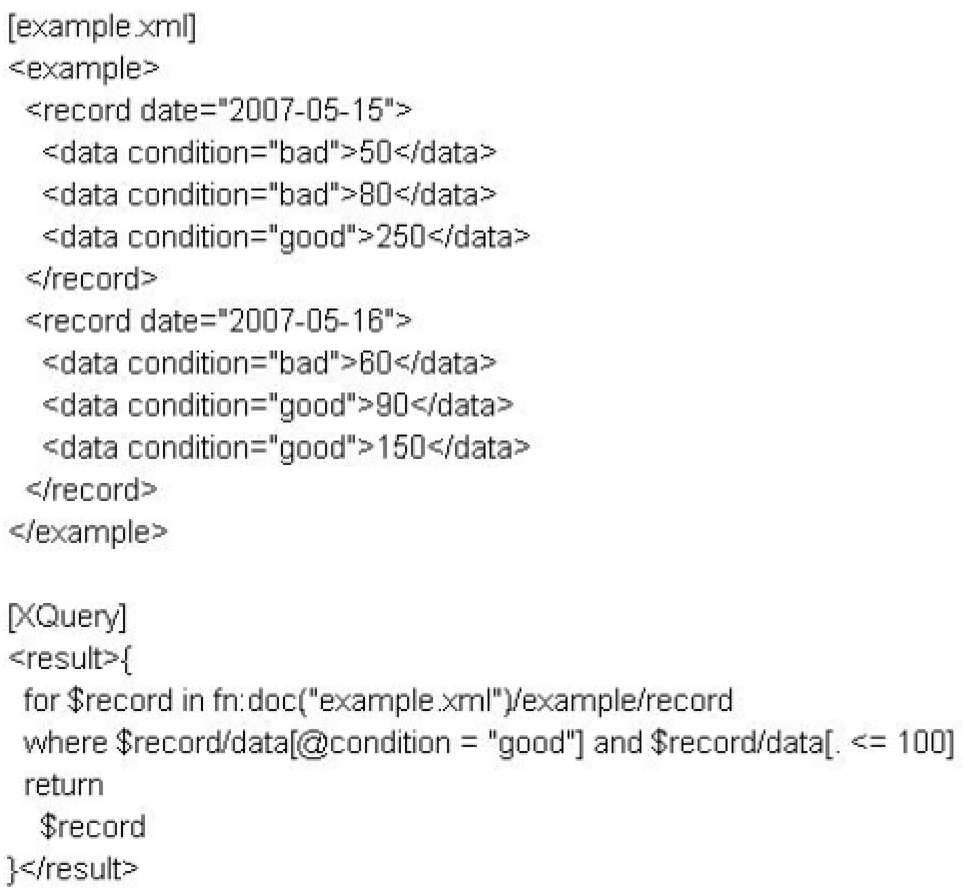
Answer:

D

Question 5

Question Type: **MultipleChoice**

See separate window.



```
[example.xml]
<example>
  <record date="2007-05-15">
    <data condition="bad">50</data>
    <data condition="bad">80</data>
    <data condition="good">250</data>
  </record>
  <record date="2007-05-16">
    <data condition="bad">60</data>
    <data condition="good">90</data>
    <data condition="good">150</data>
  </record>
</example>

[XQuery]
<result>{
  for $record in fn:doc("example.xml")/example/record
  where $record/data[@condition = "good"] and $record/data[. <= 100]
  return
    $record
}</result>
```

Assume you wish to execute a query on [example xml] (separate window) to obtain a record element that includes a data element for which the condition attribute value is "good," and for which the element value is 100 or less. Select the correct result of executing the [XQuery] (separate window). The expected result would be "C;" however, the query may not be processed as expected.

Options:

A-
```
<result/>
```

B-
```
<result>
<record date='2007-05-15'>
<data condition='bad'>50</data>
<data condition='bad'>80</data>
<data condition='good'>250</data>
</record>
</result>
```

C-
```
<result>
<record date='2007-05-16'>
<data condition='bad'>60</data>
<data condition='good'>90</data>
<data condition='good'>150</data>
</record>
</result>
```

D-
```
<result>
<record date='2007-05-15'>
<data condition='bad'>50</data>
<data condition='bad'>80</data>
<data condition='good'>250</data>
</record>
<record date='2007-05-16'>
<data condition='bad'>60</data>
```

```
<data condition='good'>90</data>
<data condition='good'> 150</data>
</record>
</result>
```

Answer:

D

Question 6

Question Type: **MultipleChoice**

Consider the execution of [XQuery] as referenced in a separate window.

Assume that the following index is used when executing this [XQuery] on a certain XMLDB.

[Index Usage Standards] (assumed)

Use only the most efficient index when an index for both $A/@type and $B/@type is created.

If only one index is created, use that index.

Do not use an index if none is created.

Assume the following circumstances regarding the statistical values related to the execution of this [XQuery].

[Statistical Value] (assumed)

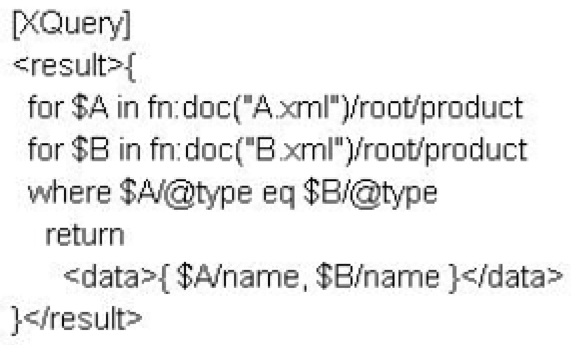
When creating an index for $A/@type, the response time when updating $A/@type data is one millisecond longeron average.

When creating an index for $B/@type. the response time when updating $B/@type data is one millisecond longeron average.

When creating an index for either $A/@type or$B/@type, or when creating an index for both, the response time of [XQuery] execution is at least two milliseconds shorter on average.

Of all of the values of $A/@type, the rate of inclusion of identical values is 1% on average. Of all of the values of $B/@type, the rate of inclusion of identical values is 2% on average. The ratio of $A/@type data update, $B/@type data update, and [XQuery] execution is 1:1:1 on average.

When the number of data items for both $A/@type and $B/@type are extremely large and almost identical, which is the most effective method for improving overall execution efficiency? Assume that this XMLDB has no other automated optimization functions than those described above. Assume that no other queries are executed.



```
[XQuery]
<result>{
  for $A in fn:doc("A.xml")/root/product
  for $B in fn:doc("B.xml")/root/product
  where $A/@type eq $B/@type
    return
      <data>{ $A/name, $B/name }</data>
}</result>
```

Options:

A- Do not create an index

B- Create an index for $A/@type

C- Create an index for $B/@type

D- Create an index for both $A/@type and $B/@type

Answer:

B

Question 7

Question Type: **MultipleChoice**

Select which of the following is not a correct description regarding static context defined by XQuery I 0.

Options:

A- Static context is information that is available during static analysis of the query expression

B- The default initial values of static context components are all set to certain values in XQuery 1.0

C- The set value of XPath 1.0 Compatibility Mode (one of the static context components) must always be 'false'

D- The default element/type namespace component is included in static context

Answer:

B

Question 8

Question Type: MultipleChoice

Assume that for [XML Document] referenced in a separate window, you wish to create an XML Schema document that defines that the value of the level element must be unique within the XML document. Which of the following correctly describes the XML Schema document?

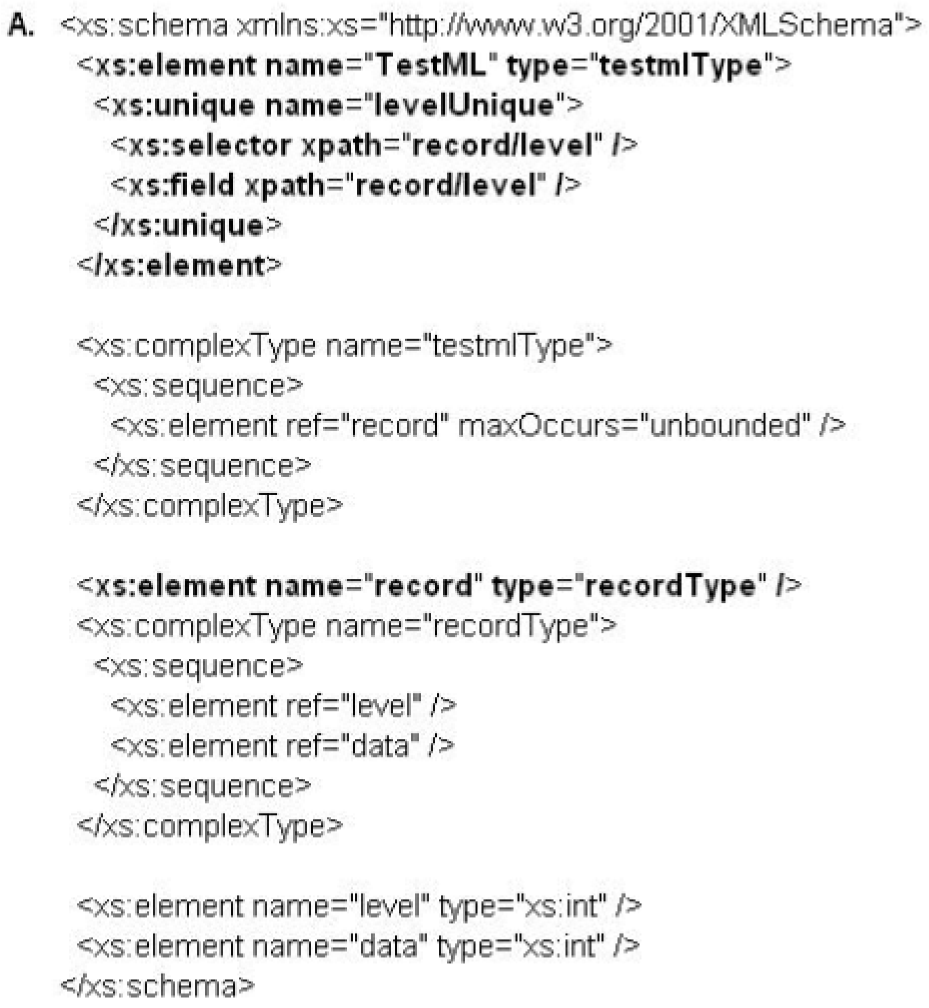
[XML Document]

1

100

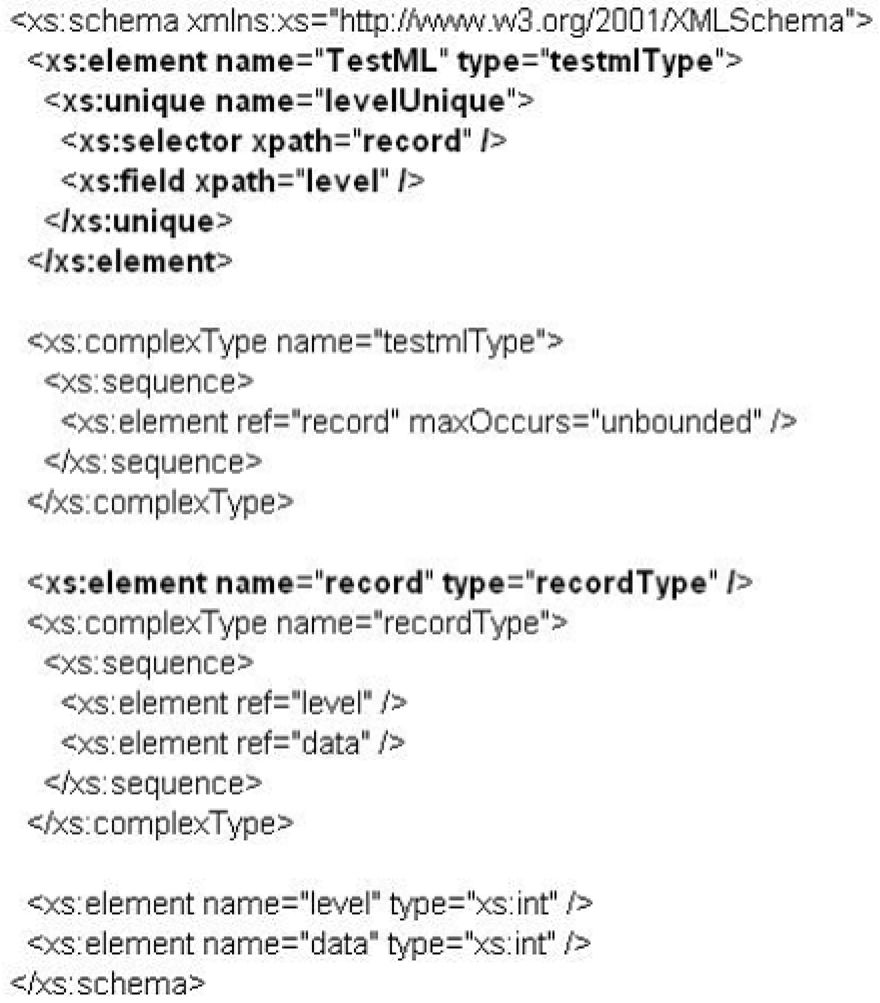
2

250



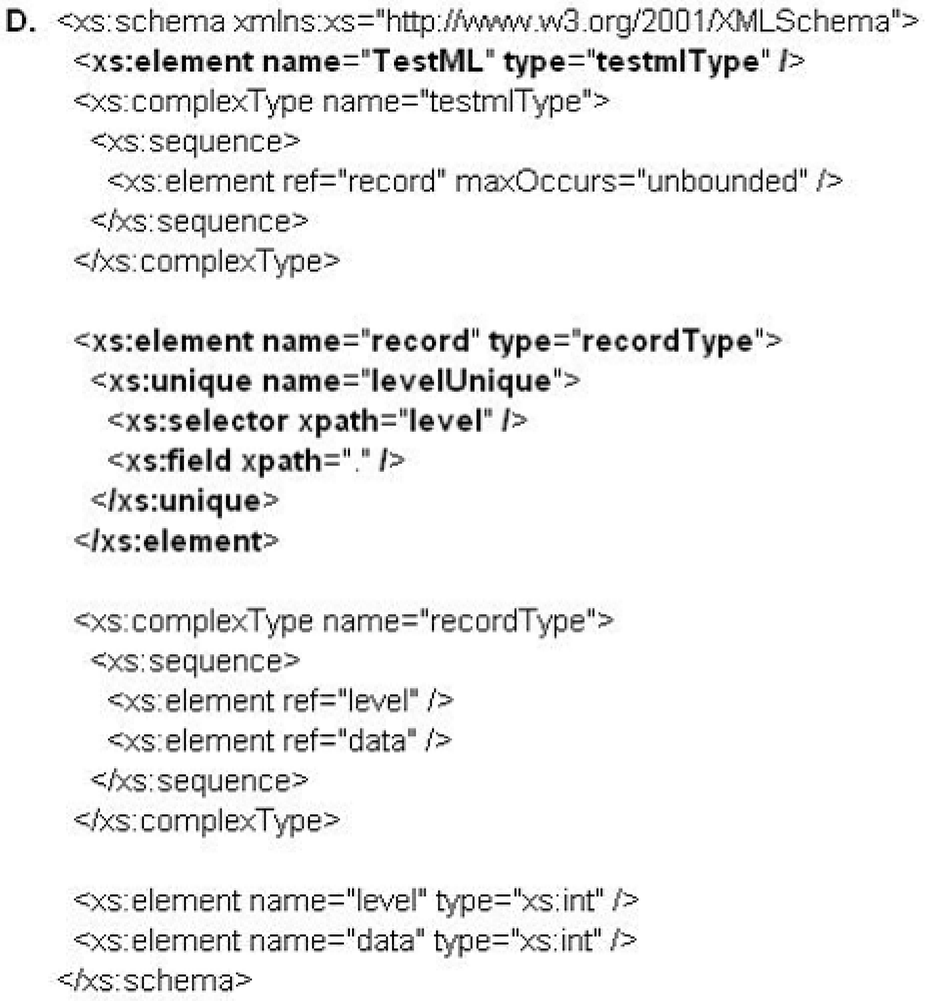
A.
```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="TestML" type="testmlType">
    <xs:unique name="levelUnique">
      <xs:selector xpath="record/level" />
      <xs:field xpath="record/level" />
    </xs:unique>
  </xs:element>

  <xs:complexType name="testmlType">
    <xs:sequence>
      <xs:element ref="record" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="record" type="recordType" />
  <xs:complexType name="recordType">
    <xs:sequence>
      <xs:element ref="level" />
      <xs:element ref="data" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="level" type="xs:int" />
  <xs:element name="data" type="xs:int" />
</xs:schema>
```

B.



```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="TestML" type="testmlType">
    <xs:unique name="levelUnique">
      <xs:selector xpath="record" />
      <xs:field xpath="level" />
    </xs:unique>
  </xs:element>

  <xs:complexType name="testmlType">
    <xs:sequence>
      <xs:element ref="record" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="record" type="recordType" />
  <xs:complexType name="recordType">
    <xs:sequence>
      <xs:element ref="level" />
      <xs:element ref="data" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="level" type="xs:int" />
  <xs:element name="data" type="xs:int" />
</xs:schema>
```

C.
```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="TestML" type="testmlType" />
  <xs:complexType name="testmlType">
    <xs:sequence>
      <xs:element ref="record" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="record" type="recordType">
    <xs:unique name="levelUnique">
      <xs:selector xpath="level" />
      <xs:field xpath="level" />
    </xs:unique>
  </xs:element>

  <xs:complexType name="recordType">
    <xs:sequence>
      <xs:element ref="level" />
      <xs:element ref="data" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="level" type="xs:int" />
  <xs:element name="data" type="xs:int" />
</xs:schema>
```



D.
```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="TestML" type="testmlType" />
  <xs:complexType name="testmlType">
    <xs:sequence>
      <xs:element ref="record" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="record" type="recordType">
    <xs:unique name="levelUnique">
      <xs:selector xpath="level" />
      <xs:field xpath="." />
    </xs:unique>
  </xs:element>

  <xs:complexType name="recordType">
    <xs:sequence>
      <xs:element ref="level" />
      <xs:element ref="data" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="level" type="xs:int" />
  <xs:element name="data" type="xs:int" />
</xs:schema>
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

B

Question 9

Question Type: MultipleChoice

Consider the structure of XML when storing date information as XML data. You must aggregate data for each calendar year from this XML data. Assume that statistical values related to the storage and retrieval of date information in a certain XMLDB is as follows:

[Date Information XML Structure A]

2007-04-OK/date>

[Date Information XML Structure B]

2QQ741

[Statistical Values] (assumed)

The search for one date node (XML structure A) or one y node (XML structure B) takes one microsecond in either case, regardless of search path.

With respect to structure A, two microseconds are required for each date node to determine whether the date element value matches the search year.

With respect to structure B, one microsecond is required for each y node to determine whether they element value is the same as the search year value.

The number of date elements at the initial state is 2000.

The number of date elements will increase with future updates; however, this is not a consideration for this question.

Insertion of a new date element under structure A takes 10 microseconds.

Insertion of a new date element under structure B takes 20 microseconds.

Do not consider any facts or conditions other than those noted above.

When the average number of new date element insertions per month is 20, and the average number of calendar date information searches per month is 10, which is the most effective XML structure in improving overall execution efficiency for insertions and searches?

Options:

A- XML Structure A

B- XML Structure B

C- Under these conditions, both are equivalent

Answer:

B

Question 10

Question Type: **MultipleChoice**

Assume that a certain XMLDB has an HTTP listener function, and can receive XML documents via HTTP protocol.

In this case, the XMLDB follows [Procedure] below to identify the character encoding method of the XML document received.

[Procedure]

(1) If the XML document character encoding method can be determined from the HTTP request header, process the XML document character encoding according to the HTTP request header information.

(2) If the XML document character encoding method cannot be determined from the HTTP request header, identify the character encoding from the XML document itself

Select which is correct as the identified character encoding resulting from the procedure above when the XMLDB receives a well-formed XML document according to [Conditions] below. Assume that the XMLDB follows RFC3023.

[Conditions]

The actual character encoding method for XML document is "UTF-16;" however, no XML declaration is specified.

The HTTP request header defines "Content-Type: text/xml." however, the charset parameter is not defined in this Content-Type header field.

Options:

A- US-ASCII

B- UTF-8

C- UTF-16

D- The character encoding method is determined from system locale, and is therefore undetermined

Answer:

A

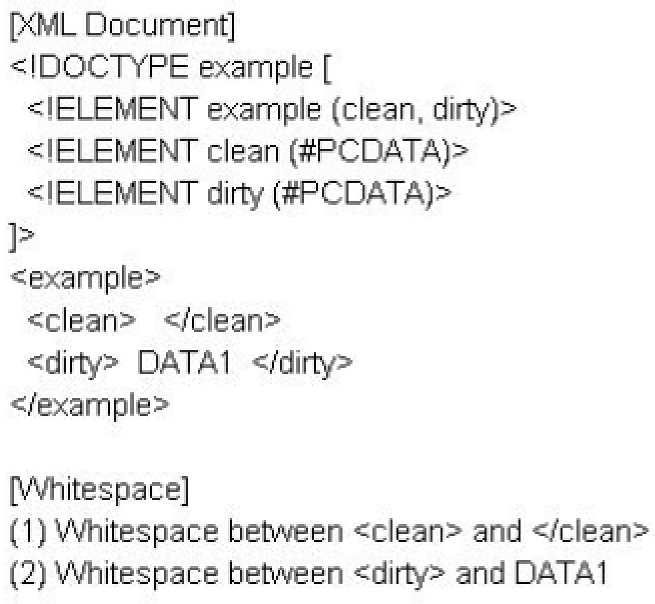
Question 11

Question Type: **MultipleChoice**

Assume that when inserting an XML document, a certain XMLDB can be configured to eliminate whitespace (ignorable white space such as line feeds, tabs, etc.) that is defined as [element content whitespace] in the XML Information Set (Infoset).

Select the correct description regarding the elimination of [Whitespace] when inserting the following [XML Document] under this whitespace elimination configuration.

Assume that a validation check is performed when the XML Document is inserted.



```
[XML Document]
<!DOCTYPE example [
  <!ELEMENT example (clean, dirty)>
  <!ELEMENT clean (#PCDATA)>
  <!ELEMENT dirty (#PCDATA)>
]>
<example>
  <clean>   </clean>
  <dirty>  DATA1  </dirty>
</example>
```

[Whitespace]
(1) Whitespace between <clean> and </clean>
(2) Whitespace between <dirty> and DATA1

Options:

A- Whitespaces of both (1) and (2) are removed

B- Whitespace of (1) is removed; whitespace of (2) is not removed

C- Whitespace of (2) is removed; whitespace of (1) is not removed

D- Whitespace of neither (1) nor (2) is removed

Answer:

D

Question 12

Question Type: **MultipleChoice**

Assume that perfect fidelity of a complete XML document is required in a certain XML document. Select the most suitable method for providing perfect XML document fidelity when storing an XML document using the following methods. Here "perfect XML document fidelity" means that an XML document retrieved from the XMLDB is exactly the same as the original XML document prior to XMLDB insertion, completely identical when compared using binary, including empty element notations, whitespace locations, etc.

Options:

A- Store the XML document as a model based on XPath 1.0

B- Store the XML document as a model based on XML Information Set (Infoset)

C- Store the XML document as a BLOB (Binary Large Object) type with no character encoding method defined

D- Store the XML document as an NCLOB (National Character Large Object) type for which UTF-8 has been defined as the character encoding method

Answer:

C

20%
DISCOUNT