Exam Number/Code: I10-003

Exam Name: XML Master Professional Database Administrator

Version: Demo

QUESTION NO: 1
Four separate operating requirements and four individual storage management methods for XML document data are listed below.
Considering the general characteristics, which individual management method ([Management Methods]) combines most optimally with which requirement ([Operating Requirements])?
[Operating Requirements]
1. Retrieve a portion of the XML document according to values in the XML document
2. Identify the XML document by unique values, and retrieve the entire XML document
3. Perform aggregation and statistical calculations of the values in the XML document
4. Continuously check the data types for the values in the XML document, and search through data using queries on the XML document
[Management Methods]
H) XML document file (text file) management via file system
I) Management via RDB (relational database), and program for storing data from an XML document into the RDB (assume the RDB does not maintain an XML document tree structure)
J) Management via XMLDB, using XML Schema definitions
K) Management via XMLDB, without using XML Schema definitions

A. H-4, I-2, J-1, K-3
B. H-1, I-2, J-4, K-3
C. H-2, I-1, J-4, K-3

Answer: A

QUESTION NO: 2
Assume that a certain XMLDB requires disk capacity in excess of the size of an XML document when storing the XML document to accommodate XML node information and other information (such as management considerations, etc.)
The following describes the capacity needed:
When eliminating ignorable whitespace in the XML document 1.5 times the XML document file size
When not eliminating ignorable whitespace in the XML document 2.0 times the XML document file size
At the initial stage, the total size of the XML document files to be stored is 1GB.
At the operating stage, repeated additions and deletions of XML documents will result in a projected disk requirement of plus or minus 10% compared to the prior year.
Assume that the disk size configured at initial stage cannot be changed for two years.
The required disk capacity will be calculated according to these conditions; however, to provide a safety margin, the decision has been made to set aside the equivalent of twice the maximum required disk capacity as calculated above.
Select the value representing the required disk capacity when ignorable whitespace is not

eliminated from the XML document.
Do not consider any facts or conditions other than those given above.

A. 3.3GB
B. 3.63GB
C. 4.4GB
D. 4.84GB

Answer: D

QUESTION NO: 3
Assume that [testmixsd] (referenced in a separate window) has been defined. Without rewriting this XML Schema Document ([testml-xsd]), create a new, separate XML Schema Document to partially change the schema definition replacing the phone element with a cellPhone element. As a result, the following [XML Document] will be valid against the new schema.

[testml.xsd]

```
<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="person" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="person" type="personType" />
  <xs:complexType name="personType">
    <xs:sequence>
      <xs:element ref="name" />
      <xs:element ref="phone" />
    </xs:sequence>
  </xs:complexType>

  <xs:element name="name" type="xs:string" />
  <xs:element name="phone" type="xs:string" />
</xs:schema>
```

Which of the following correctly descnbes the new XML Schema Document?
Assume that the XMLDB or XML parser correctly processes the XML Schema schema Location attribute.
[XML Document]
<TestML>
<person> <name>John Smith</name>
<cellPhone>000-1111-2222</cellPhone>
</person>
</TestML>

A. <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:import schemaLocation="testml.xsd" />
<xs:element name="cellPhone" type="xs:string" />
</xs:schema>
B. <xs:schema xmlns:xs="http://wvwv.w3.org/2001/XMLSchema">
<xs:include schemaLocation="testmLxsd" />
<xs:element name="cellPhone" substitutionGroup="phone" type="xs:string" />
</xs:schema>
C. <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:redefine schemaLocation-'testml.xsd">
<xs:element name="cellPhone" base="phone" type="xs:string" />
</xs:redefine>
</xs:schema>
D. This type of definition cannot be created using XML Schema

Answer: B

QUESTION NO: 4
See separate window.
Assume you wish to execute a query on [example.xml] (separate window) to obtain a record element that includes a data element having a value equal to or greater than 100 and less than 200. 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.

[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[100 <= data and data < 200]
 return
  $record
}</result>
```

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>
<recorddate="2007-05-16">
<data condition="bad">60</data>
<data condition="good">90</data>
<datacondition="good">150</data>
</record>
</result>
D. <result>
<recorddate="2007-05-15">
<data condition="bad">50</data>
<data condition="bad">80</data>
<datacondition="good">250</data>

</record>
<recorddate="2007-05-16">
<data condition="bad">60</data>
<data condition="good">90</data>
<data condition="good"> 150</data>
</record>
</result>

Answer: D

QUESTION NO: 5
Assume that you wish to create an XML Schema document against which [XML Document] (referenced in a separate window) is valid.

```
[XML Document]
<TestML xmlns="urn:xmlmaster:data">
 <division id="D001" xmlns="urn:xmlmaster:division">
  <name>Sales Department No1</name>
 </division>
 <title id="T001" xmlns="urn:xmlmaster:title">
  <name>Leader</name>
 </title>
 <person>
  <name>John Smith</name>
  <division>D001</division>
  <title>T001</title>
 </person>
 <person>
  <name>Harold Jones</name>
  <division>D001</division>
 </person>
</TestML>
```

Select which of the two answers below are correct as descriptions for (1) in the following [XML Document]. Assume that the elements with namespace "um:xmlmaster division" or "urrvxmtmaster:title' in [XML Document] have been property defined in separate XML Schemas.

[XML Schema]

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
        targetNamespace="urn:xmlmaster:data"
        xmlns:tns="urn:xmlmaster:data"
        elementFormDefault="qualified">

  <xs:element name="TestML" type="tns:testmlType"/>
  <xs:complexType name="testmlType">
      ------(1)------
  </xs:complexType>

  <xs:element name="person" type="tns:personType"/>
  <xs:complexType name="personType">
   <xs:sequence>
    <xs:element name="name" type="xs:string"/>
    <xs:element name="division" type="xs:string"/>
    <xs:element name="title" type="xs:string" minOccurs="0"/>
   </xs:sequence>
  </xs:complexType>
</xs:schema>
```

A. <xs:sequence>
<xs:any namespace="##any" processContents="lax"/>
<xs:element ref="tns:person"maxOccurs="unbounded"/>
</xs:sequence>
B. <xs:sequence>
<xs:any namespace="##other" minOccurs="0" maxOccurs="unbounded"
processContents="lax"/>
<xs:element ref="tns:person" maxOccurs="unbounded"/>
</xs:sequence>
C. <xs:sequence>
<xs:any namespace-'urn:xmlmaster: division urn:xmlmaster:title"
processContents="lax"/>
<xs:element ref="tns:person" maxOccurs="unbounded"/>
</xs:sequence>
D. <xs:choice minOccurs="0" maxOccurs="unbounded">
<xs:any namespace="urrrxmlmaster:division um:xmlmaster:title"
processContents="lax"/>
<xs:element ref="tns:person"/>
</xs:choice>

Answer: B,D

QUESTION NO: 6
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 attribute of the record element must be unique within the XML document, and further that the level attribute of the scenario element must reference the value of the level attribute of the record element.
[XML Document]

```
<TestML>
<record level="1" data="100" />
<record level="2" data="250" />
<scenario stage="A" level="1" />
<scenario stage="B" level="2" />
</TestML>
```

Select which answer correctly belongs in (1) of the [XML Schema] document below.

[XML Schema]

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

 ------(1)------

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

 <xs:element name="record" type="recordType" />
 <xs:complexType name="recordType">
  <xs:attribute name="level" type="xs:int" />
  <xs:attribute name="data" type="xs:int" />
 </xs:complexType>

 <xs:element name="scenario" type="scenarioType" />
 <xs:complexType name="scenarioType">
  <xs:attribute name="stage" type="xs:string" />
  <xs:attribute name="level" type="xs:int" />
 </xs:complexType>
</xs:schema>
```

A.
```
<xs:element name="TestML" type="testmlType">
  <xs:unique name="LEVEL">
    <xs:selector xpath="record" />
    <xs:field xpath="record/@level" />
  </xs:unique>
  <xs:ref name="levelRef" refer="LEVEL">
    <xs:selector xpath="scenario" />
    <xs:field xpath="scenario/@level" />
  </xs:ref>
</xs:element>
```

B.
```
<xs:element name="TestML" type="testmlType">
  <xs:unique name="LEVEL">
    <xs:selector xpath="record" />
    <xs:field xpath="@level" />
  </xs:unique>
  <xs:ref name="levelRef" refer="LEVEL">
    <xs:selector xpath="scenario" />
    <xs:field xpath="@level" />
  </xs:ref>
</xs:element>
```

C.
```
<xs:element name="TestML" type="testmlType">
  <xs:key name="LEVEL">
    <xs:selector xpath="record" />
    <xs:field xpath="record/@level" />
  </xs:key>
  <xs:keyref name="levelRef" refer="LEVEL">
    <xs:selector xpath="scenario" />
    <xs:field xpath="scenario/@level" />
  </xs:keyref>
</xs:element>
```

D.
```
<xs:element name="TestML" type="testmlType">
  <xs:key name="LEVEL">
    <xs:selector xpath="record" />
    <xs:field xpath="@level" />
  </xs:key>
  <xs:keyref name="levelRef" refer="LEVEL">
    <xs:selector xpath="scenario" />
    <xs:field xpath="@level" />
  </xs:keyref>
</xs:element>
```

A. Option A
B. Option B
C. Option C

D. Option D

Answer: D

QUESTION NO: 7
Select the correct result of executing the [XQuery] 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) = ("record", "name")) then
  element { fn:name($n) }
        { for $c in $n/* return local:func($c) }
 else if (fn:name($n) = "example") then
  for $c in $n/* return local:func($c)
 else ( )
};
<result>{
 local:func(fn:doc("example.xml")/example)
}</result>
```

A.
<result>
<record>
<name/>
</record>
<record>
<name/>

</record>
</result>
B.
<result>
<example>
<record>
<name/>
</record>
<record>
<name/>
</record>
</example>
</result>
C.
<result>
<record>
<name>John Smith</name>
</record>
<record>
<name>Harold Jones</name>
</record>
</result>
D.
<result>
<example>
<record>
<name>John Smith</name>
</record>
<record>
<name>Harold Jones</name>
</record>
</example>
</result>

Answer: A

QUESTION NO: 8
Select which of the following is not a correct description regarding dynamic context defined by XQuery 1.0.

A. Dynamic context is information that is available at the time the expression is evaluated
B. The dynamic context consists of all the components of the static context (default element/type namespace, etc.), and additional components (context item, etc.)

C. The value of context size (one of the dynamic context components) can be obtained using the "fn:last()" function
D. Query prolog cannot be used to set the value for any of the dynamic context components

Answer: D

QUESTION NO: 9
An [XQuery] was executed to join [eventList.xml] and [sessionUst.xml] (referenced in a separate window) and obtain the following [Expected Execution Result].
However, executing this [XQuery] resulted in an error.
Select which of the following is an appropriate example of the resulting Error Message.

[eventList.xml]
```
<eventList>
 <XMLSession>
  <schedule date="2007-05-15">
   <session sID="x999" start="00:00:00" end="00:00:00"/>
   <session sID="s050" start="13:00:00" end="13:50:00"/>
   <session sID="s051" start="14:00:00" end="14:50:00"/>
  </schedule>
 </XMLSession>
</eventList>
```

[sessionList.xml]
```
<sessionList>
 <session sID="s001">
  <title>Industry Trends</title>
 </session>
 <session sID="s050">
  <title>About XMLDB</title>
 </session>
 <session sID="s051">
  <title>About XQuery</title>
 </session>
</sessionList>
```

```
[Expected Execution Result]
<result>
  <session>About XMLDB#start=13:00:00#end=13:50:00</session>
  <session>About XQuery#start=14:00:00#end=14:50:00</session>
</result>

[XQuery]
<result>{
  for $s1 in fn:doc("eventList.xml")//session,
      $s2 in fn:doc("sessionList.xml")//session
  where $s1/@sID = $s2/@sID
  return
    <session>{ $s2/title }#start={ $s1/@start }#end={ $s1/@end }</session>
}</result>
```

A. XPST0008:An expression refers to a variable name that is not defined in the static context.
B. XPTY0018:The result of the last step in a path expression contains both nodes and atomic values.
C. XGTY0024:The content sequence in an element constructor contains an attribute node following a node that is not an attribute node.
D. XGST0054:A variable depends on itself.

Answer: C

QUESTION NO: 10
Assume the use of XML Data like [XML Data] referenced in a separate window.
When a data element is present in XML Data, there are three possibilities for b element as shown in [XML Data b Element] (separate window). No other possibilities are available.
Consider inserting any XML data into an XMLDB using the methods shown in [Operation 1] or [Operation 2].

[XML Data]

```
<example>
 <data>
  <a>peach</a>
  <b>baseball</b>
  <c>yellow</c>
 </data>
 <data>
  <a>strawberry</a>
  <b/>
  <c>red</c>
 </data>
</example>
```

[XML Data b Element]

1. The b element value is some type of character string (however, it is understood that the value is not character string stating "null")
2. The b element value is an empty string
3. The b element value is null

Select the correct answer that allows for a clear identification of the three different possibilities for [XML Data b Element].

[Operation 1]

After validating against the following [XML Schema], insert XML data (having no validation errors) into the XMLDB.

When no b element is present, the b element value is null.

When b element is present, the b element value is either 1 or 2 in [XML Data b Element].

[XML Schema]

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
 <xs:element name="example" type="EXAMPLE"/>
 <xs:complexType name="EXAMPLE">
  <xs:sequence>
   <xs:element ref="data" maxOccurs="unbounded"/>
  </xs:sequence>
 </xs:complexType>

 <xs:element name="data" type="DATA"/>
 <xs:complexType name="DATA">
  <xs:sequence>
   <xs:element name="a" type="xs:string" minOccurs="0"/>
   <xs:element name="b" type="xs:string" minOccurs="0"/>
   <xs:element name="c" type="xs:string" minOccurs="0"/>
  </xs:sequence>
 </xs:complexType>
</xs:schema>
```

[Operation 2]

After validating against the following [DTD], insert XML data (having no validation errors) into the XMLDB (do not consider the existence of a document type declaration in the XML data).

When the b element value is "null", the value of the b element is null.

When the b element value is a character string other than "null", the b element value is 1 in [XML Data b Element].

When the b element is an empty element, the b element value is 2 in [XML Data b Element].

[DTD]

```
<!ELEMENT example (data+)>
<!ELEMENT data (a?, b?, c?)>
<!ELEMENT a (#PCDATA)>
<!ELEMENT b (#PCDATA)>
<!ELEMENT c (#PCDATA)>
```

A. Under the method described in [Operation 1], the three possibilities may not be clearly identified in some circumstances

B. Under the method described in [Operation 2],the three possibilities may not be clearly identified in some circumstances

C. The methods in [Operation 1] and [Operation 2] can clearly identify the three possibilities

D. The methods in [Operation 1] and [Operation 2] cannot clearly identify the three possibilities

Answer: B