

Corrigé Examen Données Semi Structurées (du 10 /05 /2023)

Exercice 1 :

- 1 → patient+
- 2 → antecedents?
- 3 → antecedents (#PCDATA)>
- 4 →]>
- 5 → name="patients"
- 6 → <xs:sequence>
- 7 → <xs:element name="age" type=
- 8 → name="antecedents"
- 9 → </xs:element>
- 10 → </xs:complexType>

Exercice 2 :

A) Écrivez une expression XPath pour :

1. sélectionner tous les éléments `nom` dans ce fichier XML.
`//nom`
2. sélectionner tous les éléments `patient` dont le traitement est "Thérapie à l'insuline".
`//patient[traitement="Thérapie à l'insuline"]`
3. sélectionner tous les éléments `patient` dont la date de traitement est supérieure à 2022-01-01.
`//patient[date > '2022-01-01']`
4. sélectionner le nom et le traitement de tous les patients dont le diagnostic est "Hypertension".
`//patient[diagnostic='Hypertension']/(nom | traitement)`
5. sélectionner le nom du premier patient de la liste.
`//patient[1]/nom`
6. sélectionner le nom et l'âge de tous les patients de sexe masculin.
`//patient[sexe = 'Masculin']/(nom | age)`
7. sélectionner tous les éléments `patient` dont le diagnostic contient le mot "cancer".
`//patient[contains(diagnostic, 'cancer')]`
8. sélectionner tous les éléments `sexe` des patients âgés de plus de 50 ans.
`//patient[age > 50]/sexe`

B) Écrivez une expression XQuery pour :

1. récupérer tous les patients dont la pression artérielle systolique est supérieure à 140
`for $patient in /dossiers_medicaux/patient
where $patient/traitement = "Médicament pour la pression artérielle"
and $patient/pression_arterielle_systolique > 140
return $patient`

Remarque : Le fichier XML fourni ne contient pas de balise pour la pression artérielle systolique. Il faut ajouter cette information dans la structure XML pour utiliser cette requête correctement.

2. récupérer la liste des médicaments prescrits au patient Iyad Rahim

```
for $patient in /dossiers_medicaux/patient[nom = "Iyad Rahim"]  
return $patient/traitement
```

3. récupérer le nombre total de consultations pour chaque patient

```
for $patient in /patients/patient  
return <patient> <nom>{$patient/nom}</nom>  
<consultations>{count($patient/consultations/consultation)}</consultations> </patient>  
for $patient in /dossiers_medicaux/patient
```

```

return <result>
  <nom>{$patient/nom}</nom>
  <nombre_consultations>{count($patient/date)}</nombre_consultations>
</result>

```

4. récupérer tous les patients atteints d'une maladie spécifique telle que « Diabète »

```
for $patient in /dossiers_medicaux/patient[diagnostic = "Diabète"]
```

```
return $patient
```

Exercice 3 (2pts) :

- 1) Le document XML pour représenter des informations sur les réseaux informatiques peut être modélisé de la manière suivante, en utilisant des éléments pour représenter les réseaux et les périphériques :

```

<reseaux>
  <reseau id="1" nom="Réseau A">
    <peripheriques>
      <peripherique nom="Routeur" type="Routeur" adresseIP="192.168.1.1"/>
      <peripherique nom="Switch" type="Switch" adresseIP="192.168.1.2"/>
      <peripherique nom="Serveur" type="Serveur" adresseIP="192.168.1.3"/>
    </peripheriques>
  </reseau>
  <reseau id="2" nom="Réseau B">
    <peripheriques>
      <peripherique nom="Routeur" type="Routeur" adresseIP="192.168.2.1"/>
      <peripherique nom="Switch" type="Switch" adresseIP="192.168.2.2"/>
      <peripherique nom="Serveur" type="Serveur" adresseIP="192.168.2.3"/>
    </peripheriques>
  </reseau>
</reseaux>

```

Dans cet exemple, chaque réseau est représenté par un élément <reseau> qui contient un identifiant unique et un nom. Chaque réseau contient également une liste de périphériques, représentée par un élément <peripheriques>, qui contient une liste d'éléments <peripherique>. Chaque périphérique est représenté par un élément <peripherique> qui contient un nom, un type et une adresse IP.

- 2) Pour ajouter un attribut "adresse MAC" optionnel pour chaque périphérique, nous pouvons simplement ajouter un nouvel attribut "adresseMAC" à chaque élément <peripherique>. Voici le code XML modifié :

```

<reseaux>
  <reseau id="1" nom="Réseau A">
    <peripheriques>
      <peripherique nom="Routeur" type="Routeur" adresseIP="192.168.1.1" adresseMAC="00:11:22:33:44:55"/>
      <peripherique nom="Switch" type="Switch" adresseIP="192.168.1.2"/>
      <peripherique nom="Serveur" type="Serveur" adresseIP="192.168.1.3" adresseMAC="AA:BB:CC:DD:EE:FF"/>
    </peripheriques>
  </reseau>
  <reseau id="2" nom="Réseau B">
    <peripheriques>
      <peripherique nom="Routeur" type="Routeur" adresseIP="192.168.2.1" adresseMAC="11:22:33:44:55:66"/>
      <peripherique nom="Switch" type="Switch" adresseIP="192.168.2.2"/>
      <peripherique nom="Serveur" type="Serveur" adresseIP="192.168.2.3" adresseMAC="BB:CC:DD:EE:FF:00"/>
    </peripheriques>
  </reseau>
</reseaux>

```

Dans cet exemple, chaque élément <peripherique> contient maintenant un attribut "adresseMAC" optionnel qui peut être utilisé pour stocker l'adresse MAC de chaque périphérique connecté. (**Notez** que certains périphériques n'ont pas d'adresse MAC définie, car **l'attribut est optionnel**).

Partie QCM : Choisir une ou plusieurs réponses pour chaque question :

1. Qu'est-ce qu'un espace de noms dans XML ?

- a. Un moyen de donner un nom unique à un élément ou à un attribut dans un document XML.
- b. Un moyen de masquer la structure d'un document XML.
- c. Un moyen de donner un nom générique à un document XML.
- d. Un moyen de diviser un document XML en plusieurs fichiers.

2. Quelle est la syntaxe correcte pour déclarer un espace de noms dans un document XML ?

- a. `<namespace:element xmlns:namespace="http://example.com">`
- b. `<element namespace:xmlns="http://example.com">`
- c. `<element xmlns:namespace="http://example.com">`
- d. `<element xmlns:namespace"http://example.com">`

3. Quelles sont les méthodes de sélection d'éléments dans XSLT ?

- a. Par nom d'élément
- b. Par nom d'attribut
- c. Par valeur d'attribut
- d. Par parent

4. Quelle est la principale utilisation d'une DTD dans un document XML ?

- a. Décrire la structure et les contraintes du document XML.
- b. Appliquer des transformations sur un document XML.
- c. Valider la syntaxe d'un document XML.
- d. Requête un document XML.

5. Quelles sont les utilisations d'un schéma XML (XSD) dans un document XML ?

- a. Décrire la structure et les contraintes du document XML.
- b. Appliquer des transformations sur un document XML.
- c. Valider la syntaxe d'un document XML.
- d. Requête un document XML.

6. Quelle est la principale différence entre une DTD et un schéma XML (XSD) ?

- a. Une DTD est plus simple et moins expressive qu'un schéma XML (XSD).
- b. Un schéma XML (XSD) est plus simple et moins expressive qu'une DTD.
- c. Une DTD et un schéma XML (XSD) sont deux noms différents pour le même concept.
- d. Une DTD et un schéma XML (XSD) sont utilisés pour des buts différents et ne peuvent pas être comparés.

7. Quels sont les types de données primitifs définis dans XML Schema (XSD) ?

- a. string
- b. boolean
- c. date
- d. decimal

8. Quelle est la principale utilisation de l'attribut "type" dans un schéma XML (XSD) ?

- a. Définir le type de données d'un élément ou d'un attribut.
- b. Spécifier l'espace de noms d'un élément ou d'un attribut.
- c. Décrire la structure et les contraintes d'un document XML.
- d. Appliquer des transformations sur un document XML.

9. Quelle est la syntaxe correcte pour déclarer un attribut dans un schéma XML (XSD) ?

- a. `<attribute name="attributename" type="datatype" />`
- b. `<attribute attributename="name" datatype="type" />`
- c. `<attribute name="attributename" type="type" />`
- d. `<attribute attributename="attributename" type="datatype" />`

10. Quels sont les opérateurs logiques utilisables dans une expression XPath ?

- a. =
- b. and
- c. or
- d. not

11. Quelles sont les différences principales entre DOM (Document Object Model) et SAX (Simple API for XML) ?

- a. DOM est une représentation en mémoire du document XML, tandis que SAX est un traitement événementiel du document XML.
- b. DOM est un traitement événementiel du document XML, tandis que SAX est une représentation en mémoire du document XML.
- c. DOM est plus rapide que SAX pour le traitement des documents XML de grande taille.
- d. SAX prend en charge la modification en place du document XML, tandis que DOM ne le permet pas.

12. Quelle API est plus adaptée pour le traitement de documents XML volumineux ?

- a. DOM
- b. SAX
- c. Les deux sont aussi performantes pour traiter des documents XML volumineux
- d. Aucune des deux, elles sont toutes les deux trop lentes pour traiter des documents XML volumineux

13. Quelle est la principale utilisation de XSLT ?

- a. Transformer un document XML en un autre format de données.
- b. Requête un document XML pour extraire des données spécifiques.
- c. Valider la syntaxe d'un document XML.
- d. Décrire la structure et les contraintes d'un document XML.

14. Quels sont les formats de sortie possibles lors de la transformation d'un document XML avec XSLT?

- a. HTML
- b. XML
- c. PDF
- d. CSV

15. Quelle est la principale utilisation de XQuery ?

- a. Transformer un document XML en un autre format de données.
- b. Requête un document XML pour extraire des données spécifiques.
- c. Valider la syntaxe d'un document XML.
- d. Décrire la structure et les contraintes d'un document XML.

16. Quelles sont les fonctions XQuery pour la sélection d'éléments ?

- a. fn:doc
- b. fn:element
- c. fn:attribute
- d. fn:text