

## Correction type de l'Examen du module :

### Données semi-structurées

#### Questions de cours (08 pts)

- Expliquer la notion d'hyper documents et citer leurs différents composants (1 pts) :

Les hyper documents sont des documents contenant plusieurs types et plusieurs formats de documents et sont composés de trois éléments essentiels :

**1. Les noeuds :** ce sont des unités d'informations simples et indivisibles caractérisés par un contenu et des attributs comme date de création du noeud , liste des mots clés, la taille du noeud, l'auteur etc ..

**2. Les liens :** ce sont les relations entre deux noeuds ou entre deux emplacements contenus dans un même noeud. Les liens sont caractérisés eux aussi par des attributs tels que le type de liens, l'auteur de liens.. etc. Il existe deux types de liens :

- **Liens hiérarchiques (organisationnels) :** liens qui relient des fragments de texte dans un même noeud.
- **Lien de référence :** l'établissement de relation entre deux noeuds.

**3. Hypermédia :** les noeuds peuvent être des textes , images , animations, sons, ou des programmes.

- Quel est le rôle des technologies à base XML suivantes : XLink, Xpointer, Shéma XML, XSLT (2 pts) :

- XLink et XPointer : établissent liens entre les documents.
- XQuery : est un langage de requêtes sur des base de données xml.
- Shémas XML : un modèle de document, construit la grammaire du document xml.
- XSLT : permet la transformation du document xml vers d'autres formats, pour faire aussi la mise en forme.

- Pour qu'un document XML soit correct, il doit être **bien formé** et **valide**. Expliquer ces deux étapes (2 pts) :

Pour qu'un document XML soit correct, il doit d'abord être bien formé et ensuite être validé.

- La première contrainte est de nature syntaxique → concerne les règles d'écritures  
→ Le document doit respecter la syntaxe du langage XML.
- La seconde contrainte est de nature structurelle → le document doit respecter le modèle d'organisation (DTD ou Schéma XML).

- Donner les différents **types des attributs** pour les éléments XML (2 pts) :

Les différents types des attributs	<b>CDATA</b> : du texte (Character data)
	<b>ID</b> : un identifiant unique (caractère et nombre)
	<b>IDREF</b> : une référence vers un ID
	<b>IDREFS</b> : une liste des références vers des IDs (séparés par de blanc)
	<b>NMTOKEN</b> : un mot (donc pas de blanc)
	<b>NMTOKENS</b> : une liste de mots (séparés par un blanc )
	<b>Une énumération de valeurs</b> : chaque valeur est séparée par le caractère

- Donner la définition du langage **XQuery** (1 pts) :

- XQuery est un langage de requêtes pour XML qui peut être vu comme une extension de XPath 2.0 (Toute requête XPath 2.0 est aussi une requête XQuery).
- XQuery langage non XML, permet de traiter des ressources XML (fichier ou SGBD XML) pour obtenir des structures XML.
- C'est un langage fonctionnel typé, il est basé sur la manipulation de liste de noeud XML.

## Exercice 1 (6 pts)

➤ **Proposer un DTD pour que le document xml « bibliothèque » soit valide (3 pts) :**

```
<? xml version="1.0" encoding="UTF-8">
<! DOCTYPE bibliotheque [
<!ELEMENT bibliotheque ( livre+, adhérent* ) >
<!ELEMENT livre ( titre , auteur+, editeur, année?, nb_pages? ) >
<!ELEMENT titre (#PCDATA) >
<!ELEMENT auteur (#PCDATA) >
<!ELEMENT editeur (#PCDATA) >
<!ELEMENT année (#PCDATA) >
<!ELEMENT nb_pages (#PCDATA) >
<!ELEMENT adherent ( nom, prenom, date_adhésion, livres_empruntés )>
<!ELEMENT nom (#PCDATA) >
<!ELEMENT prenom (#PCDATA) >
<!ELEMENT date_adhesion (#PCDATA) >
<!ELEMENT livres_empruntés (livre_emprunté* ) >
<!ELEMENT livre_emprunté ( titre_emp , liste_auteurs_emp, editeur_emp, année_emp?,
nb_pages_emp? ) >
<!ELEMENT titre_emp (#PCDATA) >
<!ELEMENT liste_auteurs_emp (#PCDATA) >
<!ELEMENT editeur_emp (#PCDATA) >
<!ELEMENT année_emp (#PCDATA) >
<!ELEMENT nb_pages_emp (#PCDATA) >
<! ATTLIST livre référence ID #IMPLIED >
<! ATTLIST adherent id_adhérent ID #REQUIRED >
<! ATTLIST livre_emprunté référence_emp IDREF #IMPLIED >
]>
```

➤ **Le document XML contenant les 3 livres ajoutés ( 3 pts ) :**

<bibliothèque>

<livre référence="0201844524" >

< titre> XML Data Management- Native XML and XML Enabled Database Systems < / titre>

< auteur > Chaudhri Akmal < / auteur >

< auteur > Rashid Awais < / auteur >

< auteur > Zicari Roberto < / auteur >

< editeur> ADDISON WESLEY < / editeur >

< année > 2003 < / année >

< nb\_pages > 641 < / nb\_pages >

</ livre>

<livre référence="2212092067" >

< titre> XML : langage et application < / titre>

< auteur> Michard Alain < / auteur >

< editeur> ADDISON WESLEY < / editeur >

< année > 2001 < / année >

< nb\_pages > 499 < / nb\_pages >

</ livre>

<livre >

< titre> Bases de données < / titre>

< auteur > Gardarin Georges < / auteur >

< editeur> ADDISON WESLEY < / editeur >

< année > 2001 < / année >

</ livre>

```
< adhérent id_adhérent ="01" >  
< nom > < / nom >  
< prenom > < / prenom >  
< date_adhésion > < / date_adhésion >  
< livres_empruntés > < / livres_empruntés >  
< / adhérent >  
< / bibliothèque >
```

## Exercice 2 (6 pts)

➤ Donner les expressions XPath des requêtes suivantes sur le document XML « bibliothèque » :

1. Le premier auteur de chaque livre (1 pts) :  
/ bibliothèque / livre / auteur [1] ou  
// livre / auteur [ 1].
2. L'ensemble des titres des livres (1 pts) :  
/ bibliothèque / livre / titre ou  
// livre/ titre.
3. Les livres dont le nombre de pages est supérieure à 100 (1 pts) :  
/bibliothèque/ livre [nb\_pages> 100]. ou  
//livre [ nb\_pages >100]
4. Les livres dont on ne connaît pas leur année (1 pts) :  
/bibliothèque/livre [not (année)] ou  
// livre [not (année)]
5. les adhérents qui sont inscrits après 01.01.2021 (date d'adhésion) (1 pts) :  
/ bibliothèque/adhérent [date\_adhésion > 01.012021] ou  
// adhérent [date \_adhésion > 01.012021]
6. les livres empruntés par l'adhérent « omar » (prénom) (1 pts) :  
/ bibliothèque/adhérent [prenom = ' omar'] / livres\_empreintés ou  
// adhérent [prenom = ' omar'] / livres\_empreintés.