

--	--	--

Modèles et Langages Bases de Données Avancées – MU4IN801**Examen répartie 2 (2e partie) du 12 janvier 2022 – durée totale : 1 heure 30**

Les documents sont autorisés. *Les téléphones mobiles doivent être éteints et rangés dans les sacs. Le barème sur 8 points (8 questions) n'a qu'une valeur indicative.*

XML (8 pts)

On considère la DTD suivante :

```
<?xml version ="1.0" encoding="ISO-8859-1"?>
<!ELEMENT SocialNetwork (user+|post*)>
<!ELEMENT user (mail,password?, friend*)>
<!ELEMENT mail (#PCDATA)>
<!ELEMENT password (#PCDATA)>
<!ATTLIST user idU ID #REQUIRED>
<!ELEMENT friend EMPTY>
<!ATTLIST friend idF IDREF #REQUIRED>
<!ELEMENT post (#PCDATA)>
<!ATTLIST post type CDATA>
<!ATTLIST post idP ID #REQUIRED>
<!ATTLIST post author IDREF #REQUIRED>
```

On veut définir un schéma *SocialNetwork.xsd* pour modéliser les éléments contenues dans la DTD précédente. Les identifiants des utilisateurs et des posts sont des entiers.

Question 12 (1 point)

Complétez la définition de l'élément *SocialNetwork*. **Ne pas** définir le contenu des éléments *user* et *post*.

Réponse :

`<xs:element name="SocialNetwork">`

Question 13 (1 point)

Complétez la définition de l'élément *user*. On ne tient pas compte des contraintes ID/IDREF.

Réponse :

`<xs:element name = "user" type="userType" />`

Question 14 (1 point)

Complétez la définition de l'élément *post*. On ne tient pas compte des contraintes ID/IDREF.

Réponse :

`<xs:element name = "post" />`

Question 15 (1 point)

On suppose maintenant que l'attribut *type* de l'élément *post* peut avoir comme valeur seulement *image* et *text*. Donnez la nouvelle définition de l'attribut *type* en **XSchema**.

`<!ATTLIST post type (image|text)>`

Réponse :

`<xs:attribute>`

Question 16 (1 point)

Donnez les définitions des contraintes liées aux attributs *idU* de l'élément *user* et *idF* de l'élément *Friend* qui devraient être ajoutées à l'intérieur de la définition de l'élément *SocialNetwork*.

Réponse :

Exprimez en XPath les requêtes suivantes :

Question 17 ($\frac{1}{2}$ point)

Les adresses mails des utilisateurs qui n'ont aucun ami en commun avec l'utilisateur 1 (*idU=1*).

Réponse :

```
//user[not(friend/@idF = //user[@idU=1]/friend/@idF)]/mail
```

Question 18 (1 point)

Les identifiants des utilisateurs *u* qui ont publié au moins deux posts.

Réponse :

```
//post[@author = following-sibling::post/@author]/@author
```

Exprimez en XQuery la requête suivante :

Question 19 (1½ points)

Pour chaque utilisateur u , s'il existe des utilisateurs qui sont amis de ses amis et qui ont déclaré u comme ami, afficher le mail de u et le mail de ces utilisateurs (chacun de ces utilisateurs doit être affiché une seule fois). Le résultat doit satisfaire la DTD suivante :

```
<!ELEMENT result (user*)>
<!ATTLIST user mail CDATA>
<!ELEMENT user (friend+)>
<!ELEMENT friend EMPTY>
<!ATTLIST friend mail CDATA>
```

Réponse :

```
for $u in //user
  for $v in //user
    where $u/friend/@idF = $v/friend/@idF
      and $v/friend/@idF = $u/@idU
  return
  structure imbriquée du résultat attendue
```