SCIENCES SORBONNE UNIVERSITÉ

Numéro d'anonymat : | | | |

Modèles et Langages Bases de Données Avancées – MU4IN801

Examen réparti 2 (1ère 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 12 points (11 questions) n'a qu'une valeur indicative.*

Requêtes SQL3 (4 pts)

Soit donné le schéma SQL3 suivant :

```
CREATE TYPE T_ATHLETE AS OBJECT (
    NOMATH          VARCHAR2(50 Byte),
    PRENOMATH       VARCHAR2(50 Byte),
    DATENAISSANCE   DATE,
    PAYS            REF T_PAYS);

CREATE TYPE ENS_ATHLETE AS TABLE OF REF T_ATHLETE;

CREATE TYPE T_PAYS AS OBJECT (
    NOMPAYS         VARCHAR2(50 Byte),
    ATHLETES        ENS_ATHLETE);

CREATE TYPE T_RANG AS OBJECT (
    RANG        INTEGER,
    ATHLETE     REF T_ATHLETE);

CREATE TYPE ENS_RANG AS TABLE OF REF T_RANG;

CREATE TYPE T_EPREUVE AS OBJECT (
    NOMEPREUVE      VARCHAR2(50 Byte),
    SPORT           REF T_SPORT,
    RESULTATS       ENS_RANG);

CREATE OR REPLACE TYPE ENS_EPREUVE AS TABLE OF REF T_EPREUVE;

CREATE TYPE T_SPORT AS OBJECT (
    NOMSPORT        VARCHAR2(50 Byte),
    EPREUVES        ENS_EPREUVE);

CREATE TABLE LESSPORTS OF T_SPORT NESTED TABLE EPREUVES STORE AS T56;

CREATE TABLE LESATHLETES OF T_ATHLETE;

CREATE TABLE LESEPREUVES OF T_EPREUVE NESTED TABLE RESULTATS STORE AS T76;

CREATE TABLE LESPAYS OF T_PAYS NESTED TABLE ATHLETES STORE AS T09;
```

Question 1 (1 point)
Les noms des épreuves du sport 'Biathlon'.

Réponse :
en utilisant la table LESEPREUVES :

en utilisant la table LESSPORTS :

Question 2 (1 point)
Les noms des sportifs français qui ont gagné au moins une médaille (rang <=3) dans une épreuve du sport 'Biathlon'.

Réponse : en utilisant la table LESEPREUVES :

Question 3 (2 points)
Les noms des athlètes qui ont participé à au moins deux épreuves différentes du sport 'Biathlon'.

Réponse : en utilisant la table LESSPORTS :

RDF (4 pts)

Considérons une base RDF décrivant des articles de recherche. Cette base est représentée par les triplets ci-dessous qui décrivent trois types d'informations :

— des articles de recherche (sujets de type *:article*) avec leurs mots-clés (renseignés par la propriété *:motcle*), leur titre (renseigné par la propriété *:titre*), et leurs auteurs (renseignés par la propriété *:hasAuthor*) ;
— les évaluations des articles par des auteurs (renseigné par la propriété *:hasReview*), chaque évaluation dispose d'une note pouvant être négative ;
— les institutions auxquelles sont rattachés les auteurs (renseignées par la propriété *:institution*).

Triplets articles de recherche

```
art:lsd a :article ;
            :motcle "data", "system" ;
            :titre "large␣scale␣data" ;
            :hasAuthor aut:ab , aut:cd  .

art:lsd :hasReview rev:lst1 , rev:lst2 .

rev:lst1 a :review; :reviewer aut:de ; :note 0 .
rev:lst2 a :review; :reviewer aut:kn ; :note 1 .

art:dl a :article ;
            :motcle "ml", "data" ;
            :titre "deep␣learning" ;
            :hasAuthor aut:ab , aut:yl, aut:kn  .

art:dl :hasReview rev:dl1  .

rev:dl1 a :review; :reviewer aut:de ; :note 2 .

aut:ab a :author ;  :institution "inria" .
aut:cd a :author ;  :institution "oxford" .
aut:yl a :author ;  :institution "stanford" .
aut:kn a :author ;  :institution "inria" .
```

Formuler, en SPARQL, les requêtes suivantes :

Question 4 (1 point)

Le titre des articles ayant parmi leurs mots-clés, le terme 'data' et ayant une note > 1.

Réponse :

Question 5 (1 point)

Le titre des articles rédigés par au moins deux auteurs de la même institution.

Réponse :

Question 6 (1 point)

Les auteurs qui ont évalué des articles et dont toutes les évaluations ont une note >= 1.

Réponse :

Question 7 (1 point)

Pour chaque institution, le nombre d'articles écrits par un auteur de cette institution (afficher l'institution et le nombre d'articles).

Réponse :

JSON/N1QL (4 pts)

On considère un jeux de données JSON décrivant des articles scientifiques soumis pour évaluation. Ce jeu contient deux collections.

Collection articles Elle renseigne, pour chaque article identifié par idArticle, son thème, son titre, la liste de ses auteurs et une liste de mots clés. Les auteurs sont identifiés par leurs orcid, et connus par leur nom et leur institution. Cette collection est décrite par le schéma ci-dessous où le signe + indique 1 ou plusieurs occurrences.

```
{
idArticle : Number,
theme : String,
titre : String,
auteurs : [ { orcid : Number, nom : String, institution : String } + ]
motsCles : [ String + ]
}
```

Collection revues Elle rapporte, pour chaque article de la collection articles, les évaluations qui sont faites par des chercheurs anonymes. Pour chaque évaluation on connaît la note (allant de -2 à +2) et le niveau d'expertise de l'évaluateur. On connaît également la décision finale qui peut être soit acceptation soit rejet. Les champs idArticle des deux collections sont compatibles pour la jointure. Cette collection est décrite par le schéma ci-dessous.

```
{
idArticle : Number,
evaluations :  [ { note : Number, expertise: Number } + ] ,
decision : String
}
```

Formuler, en N1QL, les requêtes suivantes :

Question 8 (1 point)

Retourner l'identifiant et le titre des articles ayant reçu une décision d'acceptation et dont toutes les évaluations ont une note > 0.

Réponse :

Question 9 (1 point)

Retourner le nombre d'auteurs et le titre des articles ayant plus de 5 auteurs ainsi que le mot clé 'data'.

Réponse :

Question 10 (1 point)

Pour chaque auteur décrit par son orcid et son nom, retourner le nombre d'articles dont il est un des auteurs.

Réponse :

Question 11 (1 point)

Pour chaque article accepté, retourner son titre et la liste des notes de ses évaluations.

Réponse :

SCIENCES SORBONNE UNIVERSITÉ

Numéro d'anonymat : | | | |

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 :

Question 18 (1 point)

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

Réponse :

Exprimez en XQuery la requête suivante :

Question 19 ($1\frac{1}{2}$ 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 :