

Nom:	Prénom:	<i>page 1</i>
-------------	----------------	---------------

Module Bases de Données et Web

Examendu 21 décembre 2006

Les documents sont autorisés – Durée: 2h.

Répondre aux questions sur la feuille du sujet dans les cadres appropriés. La taille des cadres suggère celle de la réponse attendue. Utiliser le dos de la feuille précédente si la réponse débordé d'un cadre. Le barème est donné à titre indicatif. La qualité de la rédaction sera prise en compte. Ecrire à l'encre bleue ou noire. Ne pas dégrafer le sujet.

Exercice 1: Questions diverses	2pts
---------------------------------------	-------------

Question 1

Enumérez au moins deux avantages de la répartition d'une base de données sur plusieurs sites.

Question 2

Enumérez deux limites du modèle DTD par rapport au modèle XSchéma.

Question 3

Enumérez deux avantages d'un SGBD orienté objet par rapport à un langage de programmation orienté objet.

Question 4

Enumérez deux avantages du modèle XML par rapport aux modèles de données classiques, tels que le relationnel ou l'orienté objet.

Exercice 2: BDréparties	4pts
--------------------------------	-------------

La base de données d'un institut de recherche en Biologie a les schémas globaux suivants:

LABORATOIRE(codeL, nom, sigle, région, adresse, directeur)

CHERCHEUR(#chercheur, codeL, nom, prénom, spécialité, salaire, prime)

PROJET(codeP, codeL, titre, chef, budget, durée, date_début)

PARTICIPE(#chercheur, codeP, rôle, taux_participation)

L'institut est formé de trois centres de recherche situés dans trois régions Françaises : Paris, Lille, et Grenoble. Chaque centre est formé de plusieurs laboratoires. En supposant que la base de données de l'institut est répartie sur trois sites informatiques correspondant aux centres de Paris, de Lille, et de Grenoble, proposer une bonne décomposition de la base sur ces trois sites en se basant sur les hypothèses suivantes.

- Chaque centre gère ses propres données.
- L'attribut *région* de LABORATOIRE prend une des valeurs suivantes: "Paris", "Lille", "Grenoble".
- L'attribut *spécialité* de CHERCHEUR prend une des valeurs suivantes: "Virologie", "Parasitologie", "Neuroscience", "Microbiologie", "Epidémiologie", "Génétique", "Immunologie".
- codeL (resp. #chercheur, codeP) est la clé primaire de LABORATOIRE (resp. CHERCHEUR, PROJET).
- Le chef d'un projet et le directeur d'un laboratoire sont des chercheurs désignés par leurs numéros.
- Chaque chercheur est rattaché à un laboratoire donné, et peut participer à plusieurs projets de son laboratoire.

Question 1

Donner la définition des différents fragments en utilisant les opérateurs de l'algèbre relationnelle.

Question 2

Dans cette question, le centre de la région Paris ne tient également lieu de siège pour l'institut. Afin de séparer les données scientifiques des données administratives, supposons maintenant que la relation **CHERCHEUR** se répartisse sur les trois sites informatiques de Paris, de Lille, et de Grenoble comme suit :

$i \in \{ \text{Paris, Lille, Grenoble} \}$
 $p_i \in \{ (\text{région} = \text{'Paris'}), (\text{région} = \text{'Lille'}), (\text{région} = \text{'Grenoble'}) \}$

$\text{CHERCHEUR_SC}_i = \pi_{\# \text{chercheur}, \text{codeL}, \text{nom}, \text{prénom}, \text{spécialité}}(\sigma_{p_i}(\text{CHERCHEUR}))$

$\text{CHERCHEUR_ADMIN}_{\text{Paris}} = \pi_{\# \text{chercheur}, \text{nom}, \text{prénom}, \text{salaire}, \text{prime}}(\text{CHERCHEUR})$

$\text{CHERCHEUR_ADMIN}_{\text{Lille}} = \text{CHERCHEUR_ADMIN}_{\text{Grenoble}} = \emptyset$

Proposez un plan d'exécution réparti de la requête permettant de retrouver, pour chaque laboratoire, le revenu le plus élevé (salaire + prime) d'un chef de projet du laboratoire. On suppose que la requête est émise par les sièges de l'institut. Le plan proposé doit optimiser le coût de communication intersites.

Exercice 3: DTD et XML Schema

6pts

L'objectif de cet exercice est d'étudier les fichiers *XMLSchema.dtd* et *XMLSchema.xsd*, qui décrivent les éléments de XML Schema.

Question 1 . (2pts)

On demande d'écrire la DTD des éléments *xs:key* et *xs:keyref* de XML Schema.

On pourra utiliser l'entité *%XPathExpr;*, qui décrit l'ensemble des expressions XPath, pour définir les expressions XPath. On suppose cette entité déjà définie.

Question 2 (2pts)

Soit la DTD suivante, décrivant l'élément *xs:attribute* de XML Schema.

```
<!ELEMENT xs:attribute ((xs:simpleType)?)>
<!-- ATTLIST xs:attribute
      name          CDATA          #IMPLIED
      id            ID              #IMPLIED
      type          CDATA          #IMPLIED
      use           (prohibited|optional|required) #IMPLIED
      default       CDATA          #IMPLIED
      fixed         CDATA          #IMPLIED-->
```

On demande d'écrire la définition de cet élément *xs:attribute* en XML Schema, en définissant le type *TypeAttribute* indépendamment.

On suppose que l'élément *simpleType* est de type *TypeSimpleType*

Question 3 (1pt)

Quelles contraintes, sur un schéma XML Schema, impliquent les extraits suivants, du fichier XMLSchema.xsd?

1)

```
<xs:keyname="type">
  <xs:selectorxpath="xs:complexType|xs:simpleType" />
  <xs:fieldxpath="@name"/>
</xs:key>
```

2)

```
<xs:keyname="identityConstraint">
  <xs:selectorxpath="//xs:key|//xs:unique|//xs:keyref"/>
  <xs:fieldxpath="@name"/>
</xs:key>
```

Question 4 (1pt)

Dans la définition de l'élément *xs:element*, les noms d'éléments utilisés comme valeur de l'attribut *ref* doivent mentionner des éléments existants (c'est-à-dire la valeur de l'attribut *name* d'un autre élément existant).

On demande de définir cette contrainte à l'aide de *keyref*, en se basant sur la contrainte de clé appelée "element" définie ci-dessous:

```
<xs:keyname="element">
<xs:selectorxpath="xs:element"/>
<xs:fieldxpath="@name"/>
</xs:key>
```

Exercice 4: XPath et XQuery**8pts**

Soit le fichier XML Atlas.xml suivant:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE atlas SYSTEM "atlas.dtd">
<atlas>
  <pays n="p2" population="82" continent="c1">
    <nom>Allemagne</nom>
    <langue>Allemand</langue>
```

```
<frontiere pays="p1"/>
</pays>
<pays n="p1" population="60" continent="c1" >
  <nom>France</nom>
  <langue pourcentage="100">Français</langue>
  <langue pourcentage="1">Corse</langue>
  <frontiere pays="p2"/>
  <frontiere pays="p3"/>
</pays>
<pays n="p3" population="40" continent="c1">
  <nom>Espagne</nom>
  <langue pourcentage="74">Espagnol</langue>
  <langue pourcentage="17">Catalan</langue>
  <langue pourcentage="7">Galicien</langue>
  <frontiere pays="p1"/>
</pays>
<pays n="p4" population="76" continent="c2">
  <nom>Egypte</nom>
  <langue>Arabe</langue>
</pays>

<continent n="c1" nom="Europe" superficie="10"/>
<continent n="c2" nom="Afrique" superficie="30"/>

<mer n="m1" nom="Mer Mediterranee" profondeur="5120">
  <situation pays="p1"/> <situation pays="p3"/> <situation pays="p4"/>
</mer>

<montagne n="M1" nom="Alpes" altitude="4810">
  <situation pays="p1"/> <situation pays="p2"/>
</montagne>
  <montagne n="M2" nom="Cevennes" altitude="1700">
    <situation pays="p1"/>
  </montagne>
</atlas>
```

Question 1 .Exprimez en XPath les requêtes suivantes:

1. Langues qui ne sont pas parlées en Europe.

2. Nom des pays méditerranéens (ayant un accès à la mer Méditerranée)

3. Nom des pays ayant des frontières avec plus de 3 autres pays.

4. Pays de plus de 40 millions d'habitants n'ayant pas de montagnes des sommets supérieurs à 2000.

Question 2 . Expliquer en une phrase en français ce que signifient les expressions XPath suivantes, et donner le résultat de leur évaluation sur le document Atlas.xml.

1. //pays[population < 80 and position() = 2]/nom

2. //pays[population < 80][position() = 2]/nom

Question 3 . Écrire en XQuery les requêtes suivantes:

1. Donner les noms des pays qui sont voisins du pays d'identificateur p1 (n = 'p1').

Le résultat doit être:

<root>

<nom> **Allemagne** </nom>

<nom> **Espagne** </nom>

</root>

2. Donner les noms des pays dont toutes les langues sont parlées par au moins 2% de la population. Le résultat doit être:

<root>

<nom> **Espagne**</nom>

</root>

3. Donner, dans l'ordre alphabétique, tous les noms de continent, suivi chacun par les noms des pays composant le continent, dans l'ordre alphabétique également. On obtient une hiérarchie à trois niveaux suivante:

<monde>

<continentnom=" **Afrique**">

<pays> **Egypte**</pays>

</continent>

<continentnom=" **Europe**">

<pays> **Allemagne**</pays>

<pays> **Espagne**</pays>

<pays> **France**</pays>

</continent>

</monde>

4. Donner tous les noms de montagne, suivi chacun par les noms des pays dans lesquels est située la montagne. On obtient le résultat suivant:

<root>

<montagnenom=" **Alpes**">

<pays> **Allemagne**</pays>

<pays> **France**</pays>

</montagne>

<montagnenom=" **Cévennes**">

<pays> **France**</pays>

</montagne>

</root>

5. Donner la population totale, pour le fichier Atlas.xml, de chaque continent.

Le résultat doit être:

```
<root>
<continentnom=" Europe">182</continent>
<continentnom=" Afrique">76</continent>
</root>
```