

# Examen final

## Module: Base de données avancées (INF345)

Pierre SENELLART  
pierre.senellart@telecom-paristech.fr

13 février 2009

### 1 Langages et requêtes XML (11 points)

- (3 points) Pour chacune des requêtes suivantes, si une forme équivalente n'utilisant pas d'expression FLWR existe (c'est-à-dire, si c'est uniquement une *expression de chemin*), la donner. Si une telle forme n'existe pas, expliquer pourquoi.
  - for \$x in doc("bib.xml")  
return (for \$y in \$x//book where \$y/title='Databases' return \$y/author)
  - for \$x in doc("bib.xml")  
return <res>{for \$y in \$x//book[year='2000'] return \$y/title }</res>  
for \$x in doc("bib.xml")//book
  - where (for \$y in \$x/author where \$y/first='Julie' return \$y)  
return \$x/title
- (2 points) Une application de gestion de données bibliographiques est organisée comme suit. Une grande base de données XML contient chacun des documents, qui peuvent être de type varié (*livre, revue* etc.). Lorsque le document est créé dans la base, un *code* (une chaîne de caractères) lui est attribué, c'est-à-dire, un identifiant unique dans la base bibliographique. Pour chaque document, l'on enregistre : son *titre*, la liste de ses *auteurs*, la *maison d'édition*, l'*année* de publication, un ensemble de *mots-clé*, et une *description* textuelle.  
Proposer un schéma de document XML permettant de représenter ces informations (au choix, soit sous la forme d'une DTD, soit en décrivant informellement l'ensemble des éléments, leurs attributs, et leur contenu).
- (1 point) Fournir un exemple de (petit) document XML conforme au schéma proposé à la question précédente.
- (2 points) En utilisant le schéma proposé précédemment, écrire des requêtes XPath 1.0 permettant de répondre aux questions suivantes (on supposera que le nœud contextuel est le nœud document) :

- a) Quel sont les titres des livres dont la description contient le mot-clef "XML" ?
  - b) Quels sont les co-auteurs de "Jean Dupont" (*inutile de chercher à enlever les doublons*) ?
5. (3 points) Écrire une feuille de style XSLT 1.0 affichant (au choix, sous forme textuelle ou HTML) la liste des livres, avec leur titre et leurs auteurs. On veillera (si on choisit la forme textuelle) à séparer convenablement les différentes informations affichées par des espaces, ponctuations, et retour à la ligne ou (si on choisit la forme HTML) à utiliser des balises appropriées.

## 2 Recherche d'information (5 points)

On considère l'ensemble des 7 documents suivants :

- $d_1$  The Eiffel Tower is the most famous monument in Paris.
- $d_2$  If you go to Paris, do not miss the Louvre museum!
- $d_3$  As the sun rose over Paris, he walked through the old garden.
- $d_4$  The Ashmolean museum is the oldest museum in Europe.
- $d_5$  Paris in August is much quieter than Paris in June.
- $d_6$  The roses in this Covent Garden shop are famous.
- $d_7$  Paris is also famous for its nice gardens.

1. (2 points) Appliquer à ces documents les étapes de prétraitement suivantes : découpage en *tokens*, *stemming* morphologique, suppression des *mots vides* (ou *stop words*). Donner pour chaque document, l'ensemble des termes après l'ensemble de ces trois étapes. Il est inutile de donner chacune des étapes intermédiaires.
2. On considère la requête booléenne  $Q$  : « *paris AND NOT museum* ».
  - a) (0,5 point) Sans se préoccuper de leur score et classement, quel est l'ensemble des résultats de  $Q$  sur les 7 documents ci-dessus ?
  - b) (1,5 point) Proposer une manière d'affecter un score de pertinence aux résultats de  $Q$ . Justifier. Sans calcul compliqué, quel est le document parmi les 7 ci-dessus qui aurait le meilleur score ?
  - c) (1 point) On suppose qu'on a un ensemble de plusieurs millions de documents, et on cherche à connaître les  $k$  meilleurs résultats à la requête  $Q$ , de manière efficace (avec  $k$  beaucoup plus petit que le nombre total de résultats). On suppose qu'on dispose à la fois d'un index inversé, et d'un index direct qui donne, pour un terme et un document, le score de ce terme dans ce document. Proposer un algorithme simple de réponse à la requête et discuter de son efficacité.

## 3 Conception de BD répartie (4 points)

Un extrait d'une base de données répertoriant des matériels réseaux installés sur un réseau local, d'adresse IP 192.21.32.0, subdivisé en 6 sous-réseaux, est donné ci-après :

STATION (adIP, config, marque, #adIPréseau)

table STATION

adIP	config	marque	adIPréseau
192.21.32.106	Linux	Dell	192.21.32.96
192.21.32.112	MacOS	Apple	192.21.32.96
192.21.32.132	Windows	Toshiba	192.21.32.128
192.21.32.51	MacOS	Apple	192.21.32.32
192.21.32.61	Windows	HP	192.21.32.32
192.21.32.81	Windows	HP	192.21.32.64
192.21.32.113	Windows	HP	192.21.32.96
192.21.32.114	Windows	HP	192.21.32.96
192.21.32.63	Windows	Dell	192.21.32.64
192.21.32.166	Linux	Toshiba	192.21.32.160
192.21.32.245	MacOS	Apple	192.21.32.232
192.21.32.134	Windows	HP	192.21.32.128
192.21.32.168	Windows	HP	192.21.32.160
192.21.32.246	Windows	HP	192.21.32.232

On souhaite répartir cette table suivant la méthodologie de fragmentation horizontale. On observe pour cela les trois requêtes les plus fréquentes :

$R_1$  : SELECT adIP, config, marque FROM STATION WHERE adIPréseau = '192.21.32.96' ;

$R_2$  : SELECT marque FROM STATION WHERE adIPréseau = '192.21.32.96' OR adIPréseau = '192.21.32.32' ;

$R_3$  : SELECT adIP, config FROM STATION WHERE marque = 'HP' ;

On pose :

$A$  : adIPréseau = '192.21.32.96'       $B$  : adIPréseau = '192.21.32.3'       $C$  : marque = 'HP'

- (1 point) On note  $C_1$ ,  $C_2$  et  $C_3$  les conditions définies dans les clauses WHERE des requêtes  $R_1$ ,  $R_2$  et  $R_3$  respectivement. Écrire les conditions  $C_1$ ,  $C_2$  et  $C_3$  en fonction de  $A$ ,  $B$  et  $C$ .
- (2 points) On considère l'ensemble  $M_1$  contenant les prédicats des requêtes et leur négation (notée  $\neg$ ). On a  $M_1 = \{C_1, C_2, C_3, \neg C_1, \neg C_2, \neg C_3\}$ . On veut construire l'ensemble  $M_2$  des prédicats composés obtenus par conjonction des prédicats de  $M_1$ .
  - Combien de prédicats contient  $M_2$ , sans compter les prédicats dont la valeur logique est toujours « faux » ?
  - Écrire tous les prédicats de  $M_2$  pouvant servir à la fragmentation horizontale correcte de la table STATION.
- (1 point) En déduire les fragments de la table STATION. Ils seront illustrés par les occurrences de la table STATION ci-dessus pour montrer que tous les enregistrements trouvent une place unique dans les fragments.