

Données du Web : XPath/XQuery

TP en binômes à rendre le 15 octobre sur Moodle

1) XQuery : Tweets

Pour cette question, vous aurez besoin de reprendre la DTD pour les Tweets réalisée lors du premier TP. La première étape consiste à faire évoluer votre DTD dans le sens suivant.

- La DTD doit permettre d'enregistrer une collection de Tweets (et non pas un seul Tweet).
- La DTD doit également permettre l'enregistrement d'un ensemble d'utilisateurs.
- Tweets et utilisateurs seront reliés par des ID/IDREF.

Une fois terminée cette étape, créer un document XML (bien évidemment, valide par rapport à votre DTD) contenant au moins 3 utilisateurs et 5 tweets.

Donner les requêtes XQuery correspondants aux expressions suivantes et évaluer ces expressions dans le document XML pour les Tweets.

Attention : il sera peut être nécessaire d'apporter des légères modifications à votre DTD afin de rendre possibles toutes les interrogations !

1. Indiquer le nombre de tweets et d'utilisateurs dans la base.
2. Donner l'ensemble des hashtags contenus dans la base.
3. Créer une liste de paires tweet-auteur, avec chaque paire contenue dans un element `result`.
4. Pour chaque utilisateur, lister le nom de l'utilisateur et la date de tous ses tweets, le tout regroupé dans un élément `result`.
5. Lister les utilisateurs qui ont publié un tweet qui a été retwitté.
6. Pour chaque tweet, indiquer la date de ses deux premières réponses. Rajouter un element vide `<nonRetwitted/>` s'il n'a pas été retwitté.
7. Lister les utilisateurs de la plateforme en ordre alphabétique.
8. Lister les tweets contenant l'hashtag `"#I<3XML"`.
9. Trouvez le tweet le plus ancien ainsi que le plus recent.
10. Pour chaque utilisateur, indiquer l'ensemble des hashtags qu'il a utilisés dans ses Tweets.
11. Pour chaque tweet ayant des références utilisateur, retournez le tweet avec la liste des références utilisateur.
12. Déclarez la fonction `local :aReponduAuTweet`, qui, étant donné un tweet, retourne tous les utilisateurs qui ont répondu au Tweet.

2) Génération de Pages HTML via XQuery

Récupérer l'état en temps réel des tations Vélomag de Montpellier à l'adresse <https://data.montpellier3m.fr/dataset/disponibilite-des-places-velomagg-en-temps-reel> (fichier TAM_MMM_VELOMAG.xml). En alternative, la fonction XQuery `doc(url)` permet la récupération de données distantes.

Écrire un programme XQuery permettant de générer une page HTML contenant trois sections qui présentent la liste des stations Vélomag triées par :

- Ordre alphabétique.
- Capacité (indiquer le nombre total de places).
- Niveau de disponibilité. Pour ce dernier cas prévoir 3 catégories : faible (moins de 30% des vélos disponibles), moyenne (entre 30% et 60% des vélos disponibles), haute (plus de 60% des vélos disponibles).

3) Propriétés des requêtes XPath

1. Reformuler les requêtes suivantes en utilisant exclusivement les axes `child`, `descendant`, `descendant-or-self`, `following` et `following-sibling`
 - `//d/preceding-sibling::c`
 - `//c/a/preceding-sibling::a/preceding::e`
 - `//d[parent::b/c]`
 - `/r/b/..//*/../preceding::d`
 - `//a/ancestor::c/child::d/parent::e`
 - `//c[preceding::d]`
2. Reformuler les requêtes `//a/following::b` et `//a/preceding::b` en utilisant les axes `descendant-or-self`, `ancestor`, `following-sibling` et `preceding-sibling`.
3. Pour chaque requête définie aux points 1 et 2, proposer un document XML pour lequel la réponse à la requête n'est pas vide, sinon expliquer pourquoi un tel document n'existe pas.

4) L'égalité dans XQuery

1. Soient X, Y, Z des séquences d'éléments XML. Est-il vrai que, dans le cadre du langage XPath, si $X = Y$ et $Y = Z$ alors $X = Z$? Est-ce le cas pour XQuery?
2. Donner une fonction XQuery qui renvoie vrai si et seulement si deux séquences sont identiques. Pour simplifier, nous ne considérons pas les attributs (mais nous considérerons bien l'ordre des éléments).