C'est la suite du TP5. On continue à travailler sur la base de données factbook.

IMPORTANT : la version de BaseX qui est installée actuellement par la distribution Debian est très buggée. Voir Version 9.0.1 critical bug – il reste d'autres bugs non corrigés. Il faut télécharger la dernière version de BaseX : Version 9.6.4. Cliquez sur BaseX964.jar (4Mo). Il suffit de double-cliquer le jar obtenu pour lancer BaseX.

- Ensuite, quand vous relancez BaseX, il faut réactiver la base *factbook*. Ouvrez le menu `Base de données/Administrer` et double-cliquez à gauche sur `factbook`. Ça doit réafficher toutes les données dans la zone des résultats.
- Téléchargez `ReponsesTP5b.xq`, vos réponses seront à écrire dedans. Ce fichier sera à déposer sur Moodle dans la partie 2 du TP.
- Quand vous travaillez sur une réponse, `(:`commentez les autres`:)`

IMPORTANT : comme pour la première partie, ne changez ni le nom, ni la structure du fichier `ReponsesTP5b.xq`. Vous devez seulement placer vos réponses en dessous des noms des questions, avec une ligne vide entre chaque question. Ne laissez pas le message « Mettez votre réponse ici» si vous répondez à la question, mettez votre réponse exactement à sa place. Laissez les commentaires `(:` et `:)` seuls sur leur ligne autour de votre réponse. Ne mettez aucune ligne vide dans le bloc de votre réponse. Si vous proposez plusieurs réponses, alors faites un bloc entre `(:` et `:)` pour chaque, sans mettre de ligne vide entre eux.

Comme toujours, faites ce que vous pouvez comme vous pouvez. Pensez à sauvegarder très régulièrement.

5. Exercices XQuery sur `factbook.xml`

5.1. Villes ayant le même nom

Écrire une requête XQuery qui affiche les noms des villes qui ont le même nom, c'est à dire les noms de villes qu'on retrouve au moins en double, ainsi que leurs différents pays. Voici le résultat attendu :

```
<identique nom="Paris">
  <pays>France</pays>
  <pays>USA</pays>
</identique>
<identique nom="Alexandria">
  <pays>Romania</pays>
  <pays>USA</pays>
  <pays>Egypt</pays>
</identique>
...
<identique nom="Arlington">
  <pays>USA</pays>
  <pays>USA</pays>
</identique>
```

Voici comment faire cette requête :

- Il y a d'abord une recherche des noms de villes. On ne garde qu'un seul exemplaire de chacun. Vous remarquerez que ce calcul est très lent. Il y a 3125 villes dans le fichier.
- Pour chaque nom de ville, il faut compter les éléments `<city>` qui ont ce même nom, ne garder que les noms associés à au moins deux villes.
- Enfin, il faut retrouver les noms des pays qui contiennent ces villes. Parfois c'est le même pays qui possède deux villes du même nom. Concernant le nom du pays, certains comme les USA en ont deux, mais il ne faut garder que le premier nom du pays. Attention, dans certains pays, les villes ne sont pas directement sous la balise `<country>` mais dans une balise `<province>`.

5.2. Langue la plus parlée

Écrire une requête XQuery qui affiche le nom de la langue qui est parlée dans le plus grand nombre de pays, ainsi que la liste de ces pays. Le fichier factbook est très incomplet, les résultats seront également très incomplets :

```
<langue nombre="19" nom="English">
  <pays>Papua New Guinea</pays>
  <pays>Anguilla</pays>
  <pays>Antigua and Barbuda</pays>
  ...
  <pays>Liberia</pays>
  <pays>Namibia</pays>
</langue>
```

Voici comment faire cette requête :

- C'est l'élément `<languages>` de `<country>` qui renseigne sur les langues parlées, peu importe le pourcentage. Mettre la liste des langues sans doublons dans une variable.
- Relire l'exemple 10 de la partie 1 du TP où il est question de trouver le genre d'arbre le plus représenté dans les données, alors qu'ici, il est question d'une langue la plus parlée.
- Il faut compter combien de pays parlent chaque langue et garder la valeur maximale. Ici c'est 19.
- Ensuite, on recommence une boucle sur les langues et du nombre de pays où on les parle, en ne gardant que la langue mentionnée 19 fois.
- Il reste à produire les noms des pays concernés.

5.3. Classement des villes hébergeant au moins quatre organisations

Écrire une requête XQuery qui affiche les noms des villes qui hébergent au moins quatre organisations. Il faut les classer dans l'ordre décroissant du nombre et pour chacune, afficher la liste des organisations en question, avec abbréviation et nom entier. Voici le résultat attendu :

```
<ville nom="Geneva">
  <org abbrev="ECE">Economic Commission for Europe</org>
  <org abbrev="EFTA">European Free Trade Association</org>
  <org abbrev="CERN">European Organization for Nuclear Research</org>
  <org abbrev="G-15">Group of 15</org>
```

```
  ...
  <org abbrev="WIPO">World Intellectual Property Organization</org>
  <org abbrev="WMO">World Meteorological Organization</org>
</ville>
<ville nom="Brussels">
  <org abbrev="ACP">African, Caribbean, and Pacific Countries</org>
  <org abbrev="Benelux">Benelux Economic Union</org>
  <org abbrev="CCC">Customs Cooperation Council</org>
  ...
```

Voici comment faire cette requête :

- L'information se trouve dans l'attribut `headq` des éléments `organization`. Cet attribut indique la ville hébergeant l'organisation.
- Le principe est de parcourir ces attributs, une seule fois chacun, et pour chacun de produire un élément `<ville>` regroupant toutes les informations demandées.
- Il y a une particularité avec Bruxelles. Cette ville a deux noms. Il faudra choisir soit le premier, soit les deux noms `<ville nom="Brussels Agglomeration">`.

5.4. Surface relative des lacs par pays

Écrire une requête XQuery qui calcule la surface totale relative (en %) des lacs de chaque pays et qui n'affiche que les pays dépassant 0%, classés par surface relative décroissante. Voici le résultat attendu :

```
<pays nom="Kenya">13.13 %</pays>
<pays nom="Tanzania">10.79 %</pays>
<pays nom="Zambia">6.63 %</pays>
<pays nom="Mozambique">4.19 %</pays>
<pays nom="Switzerland">3.23 %</pays>
...
<pays nom="Venezuela">0.39 %</pays>
```

Voici comment faire cette requête :

- Le principe général est une boucle sur les pays. On doit additionner les surfaces de tous leurs lacs et faire le rapport entre cette somme et la surface totale du pays.
- Les informations sont dans `<country total_area="...">`, ainsi que dans les éléments `<lake area="..."><located country="id"/>`.
- Pour les calculs, il faut utiliser la fonction `sum(...)`. La division ne s'écrit pas avec le signe `/` mais l'opérateur `div`. Ex: `15 div 7` = 2.142857 (NB: la division entière s'écrit `idiv`). Comme vous avez un pourcentage à calculer, n'oubliez pas de multiplier par 100. Pour arrondir à deux chiffres après la virgule, utilisez la fonction `round(nombre, chiffres)`.

5.5. Population totale par continent

Écrire une requête XQuery qui calcule la population totale de chaque continent en additionnant les populations des pays qu'il regroupe, et en tenant compte du pourcentage d'appartenance de

chaque pays aux continents (mais sans vérifier que la somme des pourcentages d'un pays vaut 100). Voici le résultat attendu :

```
<continent name="Europe">630969958</continent>
<continent name="Asia">3594984260</continent>
<continent name="America">783861486</continent>
<continent name="Australia/Oceania">29729307</continent>
<continent name="Africa">724680819</continent>
```

Voici comment faire cette requête :

- Il y a une boucle sur les continents où on calcule la population totale.
- Ensuite, l'information se trouve dans les sous-éléments `<encompassed continent="..." percentage="..."/>` puis dans `<country population="...">`. Le but est d'additionner les produits $population * percentage/100$ concernant un même continent. La division se fera avec `idiv` pour rester avec des entiers.
- Il est possible que les nombres ne soient pas affichés sous forme d'entiers, mais en notation scientifique. Utilisez alors `format-number($poptot, '9')`.

5.6. Modifier la population de Londres

Écrire une requête XQuery qui modifie la valeur de la population de Londres en Angleterre, sa valeur estimée au 1er janvier 2022 est de 9540576 (source internet). Attention, il y a un piège avec une autre ville qui s'appelle Londres et qui n'est pas concernée. Modifiez la population ainsi que l'année. Voir le dernier exemple de modifications dans le tutoriel de la première partie du TP pour savoir comment faire plusieurs changements ensemble.

5.7. Supprimer les lacs sans pays

Écrire une requête XQuery qui supprime les `<lake>` qui n'ont aucun élément `<located>`.

- Faites d'abord afficher les éléments concernés pour être sûr de supprimer les bons.
- Pour réinitialiser la base, il suffit de la re-créer.

5.8. Ajouter la ville de Vannes

Écrire une requête XQuery qui ajoute un élément décrivant Vannes : province=Bretagne, longitude=-2.7603, latitude=47.5978, population en 2019=53719.

- Cet élément doit avoir la même structure que les éléments existants : il doit être validable par la DTD.
- Les informations telles que les identifiants doivent, de préférence, être recherchées par calcul à partir des noms en clair, et pas par affectation ad-hoc. Par contre, l'identifiant de la nouvelle ville peut être choisi arbitrairement, ex: `id_vannes`.

6. Travail à rendre

Déposez seulement votre fichier `ReponsesTP5b.xq` (tel quel, non compressé) sur Moodle, dans la page de cours dédiée L4IN121T Formats et traitements de données internet.