TD : Requêtes à une et plusieurs tables

1 Propagation d'épidémies (Extrait de Mines 2016)

Pour suivre la propagation des épidémies, de nombreuses données sont recueillies par les institutions internationales comme l'OMS. Par exemple, pour le paludisme, on dispose de deux tables :

— La table `palu` recense le nombre de nouveaux cas confirmés et le nombre de décès liés au paludisme ; certaines lignes de cette table sont données en exemple (on précise que `iso` est un identifiant unique pour chaque pays) :

nom	iso	annee	cas	deces
Bresil	BR	2009	309 316	85
Bresil	BR	2010	334 667	76
Kenya	KE	2010	898 531	26 017
Mali	ML	2011	307 035	2 128
Ouganda	UG	2010	1 581 160	8 431
...				

— la table `demographie` recense la population totale de chaque pays ; certaines lignes de cette table sont données en exemple :

pays	periode	pop
BR	2009	193 020 000
BR	2010	194 946 000
KE	2010	40 909 000
ML	2011	14 417 000
UG	2010	33 987 000
...		

1. Au vu des données présentées dans la table `palu`, parmi les attributs `nom, iso` et `annee`, quels attributs peuvent servir de clé primaire ? Un couple d'attributs pourrait-il servir de clé primaire ? (on considère qu'une clé primaire peut posséder plusieurs attributs). Si oui, en préciser un.
2. Écrire une requête en langage SQL qui récupère depuis la table `palu` toutes les données de l'année 2010 qui correspondent à des pays où le nombre de décès dus au paludisme est supérieur ou égal à 1 000.

On appelle *taux d'incidence d'une épidémie* le rapport du nombre de nouveaux cas pendant une période donnée sur la taille de la population-cible pendant la même période. Il s'exprimer généralement en « nombre de nouveaux cas pour 100 000 personnes par année ». Il s'agit d'un des critères les plus importants pour évaluer la fréquence et la vitesse d'apparition d'une épidémie.

3. Écrire une requête en langage SQL qui détermine le taux d'incidence du paludisme en 2011 pour les différents pays de la table `palu`.
4. Écrire une requête en langage SQL permettant de déterminer le nom du pays ayant eu le deuxième plus grand nombre de nouveaux cas de paludisme en 2010 (on pourra supposer qu'il n'y a pas de pays *ex æquo* pour les nombres de cas).

2 Analyse de performance de code (Adapté de Mines 2019)

Au cours du développement de fonctions nécessaires à la manipulation des nombres premiers on s'aperçoit que le choix des algorithmes pour évaluer chaque fonction est primordial pour garantir des performances acceptables. On souhaite donc mener des tests à grande échelle pour évaluer les performances réelles du code qui a été développé. Pour ce faire on effectue un grand nombre de tests sur une multitude d'ordinateurs. Les données sont ensuite centralisées dans une base de données composée de deux tables.

La première table est `ordinateurs` et permet de stocker des informations sur les ordinateurs utilisés pour les tests. Ses attributs sont :

— nom TEXT, clé primaire, le nom de l'ordinateur.

— gflops INTEGER la puissance de l'ordinateur en milliards d'opérations flottantes par seconde.

— ram INTEGER la quantité de mémoire vive de l'ordinateur en Go.

Exemple du contenu de cette table :

```
nom               gflops      ram
----------        ----------  ----------
nyarlathotep114   69          32
nyarlathotep119   137         32
...
shubniggurath42   133         16
azathoth137       85          8
```

La seconde table est `fonctions` et stocke les informations sur les tests effectués pour différentes fonctions en cours de développement. Ses attributs sont :

— `id INTEGER` l'identifiant du test effectué.

— `nom TEXT` le nom de la fonction testée (par exemple li, Ei, etc).

— `algorithme TEXT` le nom de l'algorithme qui permet le calcul de la fonction testée (par exemple BBS si on teste une fonction de génération de nombres aléatoires).

— `teste_sur TEXT` le nom du PC sur lequel le test a été effectué.

— `temps_exec INTEGER` le temps d'exécution du test en millisecondes.

Exemple du contenu de cette table :

```
id          nom          algorithme  teste_sur         temps_exec
----------  ----------   ----------  ----------        ----------
1           li           rectangles  nyarlathotep165   2638
2           li           rectangles  shubniggurath28   736
3           li           trapezes    nyarlathotep165   4842
...
2154        Ei           puiseux     nyarlathotep145   2766
2155        aleatoire    BBS         azathoth145       524
```

5. Quelle est l'utilité de l'attribut `teste_sur` dans la table `fonctions` ?
6. Ecrire une requête `SQL` permettant de connaître le nombre d'ordinateurs disponibles et leur quantité moyenne de mémoire vive.
7. Ecrire une requête `SQL` permettant, pour la fonction nommée `Ei`, de trier les résultats des tests du plus lent au plus rapide. Pour chaque test retenir le nom de l'algorithme utilisé, le nom du pc sur lequel il a été effectué et la puissance du PC.
8. Ecrire une requête `SQL` permettant d'extraire les noms des PC sur lesquels l'algorithme `rectangles` n'a pas été testé pour la fonction nommée `li`.