



Devoir surveillé n°2

Mercredi 18 Décembre
Durée : 2 heures

La qualité de la présentation (notamment de la lisibilité) et la justification ou l'explication des commandes utilisées sont des éléments importants pour l'évaluation de la copie.

Exercice 1

Le système *Hawk-Eye* permet d'enregistrer en temps réel les trajectoires des balles lors d'un tournoi de tennis. Ces données sont stockées dans une base de données constituée de deux tables.

La table **MATCHS** contient les différents matchs du tournoi avec les attributs :

- ✗ **id** : identifiant de type entier, clé primaire;
- ✗ **numero** : numéro du match dans la planification du tournoi;
- ✗ **date** : date où le match s'est déroulé;
- ✗ **joueur1** : nom du premier joueur;
- ✗ **joueur2** : nom du deuxième joueur (les joueurs 1 et 2 sont rangés par ordre alphabétique) ;
- ✗ autres attributs non détaillés...

La table **POINTS** contient des entités correspondant aux points joués lors d'un match. Elle contient les attributs :

- ✗ **id** : identifiant de type entier, clé primaire;
- ✗ **mid** : identifiant du match correspondant à la définition de type entier;
- ✗ **nombre** : nombre d'échanges de type entier;
- ✗ **fichier** : nom du fichier image de la trajectoire (stockée) correspondant au point;
- ✗ autres attributs non détaillés...

1. Écrire une requête SQL permettant d'afficher les identifiants des matchs joués par Federer, joueur pris pour exemple.
2. Écrire une requête SQL permettant d'afficher le nombre d'échanges maximum lors du match dont l'identifiant est `mid=4`.
3. Écrire une requête SQL permettant de récupérer le nom des fichiers de toutes les images qui correspondent aux matchs opposant Federer et Djokovic.

Exercice 2

Pour suivre la propagation des épidémies, de nombreuses données sont recueillies par les institutions internationales comme l'O.M.S. 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	309316	85
Bresil	BR	2010	334667	76
Kenya	KE	2010	898531	26017
Mali	ML	2011	307035	2128
Ouganda	UG	2010	1581160	8431
...				

- ✗ La table **demographie** recense la population totale de chaque pays; certaines lignes de cette table sont données en exemple :

pays	periode	pop
BR	2009	193020000
BR	2010	194946000
KE	2010	4090900
ML	2011	14417000
UG	2010	33987000
..		

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 1000.

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'exprime en général en *nombre de nouveaux cas pour 100000 personnes par année*.

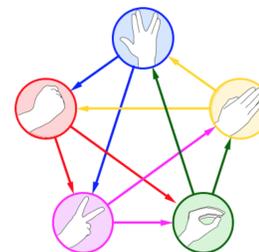
Il s'agit d'un des critères les plus importants pour évaluer la fréquence et la vitesse d'apparition d'une épidémie.

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 ex-æquo pour les nombres de cas).

Exercice 3

Inutile de rappeler les règles de *Shifumi* (ou Pierre-Feuille-Ciseaux). Ce jeu bien connu connaît aussi des variantes, comme celle (présentée ci-après) popularisée par la série télévisée *The Big Bang Theory*.

Un *graphe de Shifumi* est un graphe censé représenter les règles d'une confrontation. Les sommets sont les *figures* possible. Il y a un arc sortant d'un sommet vers un autre si la figure du sommet dont on sort l'emporte sur celle du sommet où on entre.



1. Implémenter en Python, à l'aide d'un dictionnaire le graphe du *shifumi* classique. Les clés du dictionnaire sont les sommets du graphe, et la valeur de chaque clé la liste des sommets qui lui sont adjacents.

2. Écrire une fonction Python d'en tête `defconfrontation(G, cle1, cle2)` : qui prend en argument un graphe G de shifumi, deux clés représentant les figures qui s'opposent selon les règles de ce graphe, et qui renvoie 1, -1 ou 0 selon que la `cle1` gagne contre la `cle2`, qu'elle perd, ou qu'il y a égalité.
3. Écrire une fonction Python d'en tête `def shifumi(G, nb_victoires)` : qui simule un duel entre l'utilisateur et l'ordinateur selon les règles suivantes :
 - ✗ Les figures et les règles du shifumi sont modélisées par le graphe G ;
 - ✗ Le vainqueur du duel est le premier à remporter `nb_victoires` confrontations
 - ✗ À chaque nouvelle confrontation, le joueur choisit la figure qu'il veut parmi celles disponibles (on pourra utiliser la commande `input()`) et l'ordinateur choisit sa figure aléatoirement de manière équiprobable sur l'ensemble des figures du graphe.
On rappelle que la commande `rd.randint(a,b)` renvoie un nombre entier aléatoire entre a et $b - 1$, choisi avec équiprobabilité.
 - ✗ Durant chaque confrontation, le programme affiche le choix de figure de l'ordinateur et le score temporaire résultat de cette confrontation ;
 - ✗ À l'issue du duel, l'ordinateur affiche 'Victoire' ou 'Défaite' ainsi que le score final.
4. Proposer un intitulé qui a permis de générer l'image ci-dessous par intelligence artificielle.

