

TP13 INTRODUCTION AUX BASES DE DONNÉES

Partie I

Installation de SQLite Manager pour Firefox

Pour installer le gestionnaire de base de données intégré à Firefox, allez dans « Outils de développement » puis « Modules complémentaires¹ ». Recherchez parmi les modules « sqlite manager » et demandez d'installer le premier résultat². Firefox vous signale que l'extension sera installée au prochain redémarrage, demandez-lui donc de « Redémarrer maintenant ».

Une fois le redémarrage effectué, si vous retournez dans le menu « Outils de développement », une ligne supplémentaire « SQLite Manager » est disponible. Sélectionnez-là et une nouvelle fenêtre « Gestionnaire SQLite » devrait s'ouvrir. Vous pouvez à présent ouvrir une base de données³ et la parcourir. N'hésitez pas à demander de l'aide si vous ne trouvez pas où exécuter le code SQL ou comment naviguer dans la base.

Pour chacune des sections suivantes, vous pourrez trouver la « bonne » requête en jouant avec SQLite Manager pour la recopier à l'endroit adéquat (sous forme de chaîne de caractères) dans le fichier TP13_intro_BDD.py afin que les tests automatiques puissent fonctionner⁴.

Partie II

Jeux de triangles

La base `triangles.sqlite` est constituée d'une seule table dont le schéma relationnel est le suivant:

```
TRIANGLE (idt:int, ab:int, bc:int, ac:int)
```

Chaque ligne représente un triangle défini par la taille de ses trois côtés (AB, AC et BC).

1. Lancer (successivement !) les requêtes suivantes en réfléchissant à leur signification/objectif:

```
SELECT COUNT(*) FROM triangles ;  
SELECT * FROM triangles WHERE ab+ac+bc=100 ;  
SELECT ab*ac*bc FROM triangles WHERE ab+ac+bc>=100 ;
```

2. Déterminer à l'aide de requêtes SQL

- a) la plus petite valeur des produits $AB \times AC \times BC$ pour les triangles ABC de périmètre supérieur ou égal à 100 ;
- b) les longueurs AB, AC et BC correspondants au(x) triangle(s) pour le(s)quel(s) le minimum précédent est atteint ;
- c) tous les triangles rectangles en A ;
- d) le nombre de tels triangles ;
- e) le maximum des périmètres des triangles rectangles en A ;
- f) tous les triangles équilatéraux ;
- g) tous les triangles tels que $\frac{AB + AC + BC}{3} = 42$.

NB: quand on dit « tous les triangles », on utilisera l'étoile pour récupérer toutes les colonnes disponibles dans l'ordre par défaut de la table.

¹Ou raccourci clavier Ctrl-Shift-A

²À la dernière vérification, c'était « SQLite Manager 0.8.1 ».

³À l'aide de l'icône représentant l'ouverture d'un dossier.

⁴Si vous êtes sûr de votre coup, vous pouvez aussi directement l'écrire dans le fichier python...

Partie III

Communes, départements et régions

1. Ouvrir la BDD `communes_departements_regions.sqlite` et écrire *sur papier* les schémas relationnels des tables qu'elle contient.
2. Lancer la requête

```
SELECT C.nom, D.nom FROM communes AS C JOIN departements AS D ON C.dep = D.id ;
```

3. En s'inspirant du modèle précédent, donner la liste de toutes les communes avec pour chacune son département, sa région et sa population.
4. Trier la liste précédente par ordre décroissant de population.
5. Déterminer le rang de la ville de Strasbourg dans ce classement.
6. Faire de même⁵ avec le Bas-Rhin pour la population des départements.
7. Et l'Alsace par rapport aux régions tant qu'à faire⁶ !
8. Donner la liste des communes (nom et population) dont le nom commence par **Pa** et se finissant par **is**. On pourra pour cela utiliser le mot-clé **LIKE** (et Google pour savoir comment l'utiliser).
9. Déterminer les communes (nom et population) qui ont strictement moins d'habitants que de lettres dans leur nom⁷.

Partie IV

Prénoms parisiens

La base `prenoms_paris.sqlite` contient les prénoms enregistrés à l'état civil de Paris depuis 2004 jusqu'en 2013.

1. Donner le schéma relationnel des différentes tables de cette base.
2. Pour chacune des requêtes SQL suivantes, donner la traduction « en français » et vérifier la vraisemblance du résultat depuis SQLite Manager.

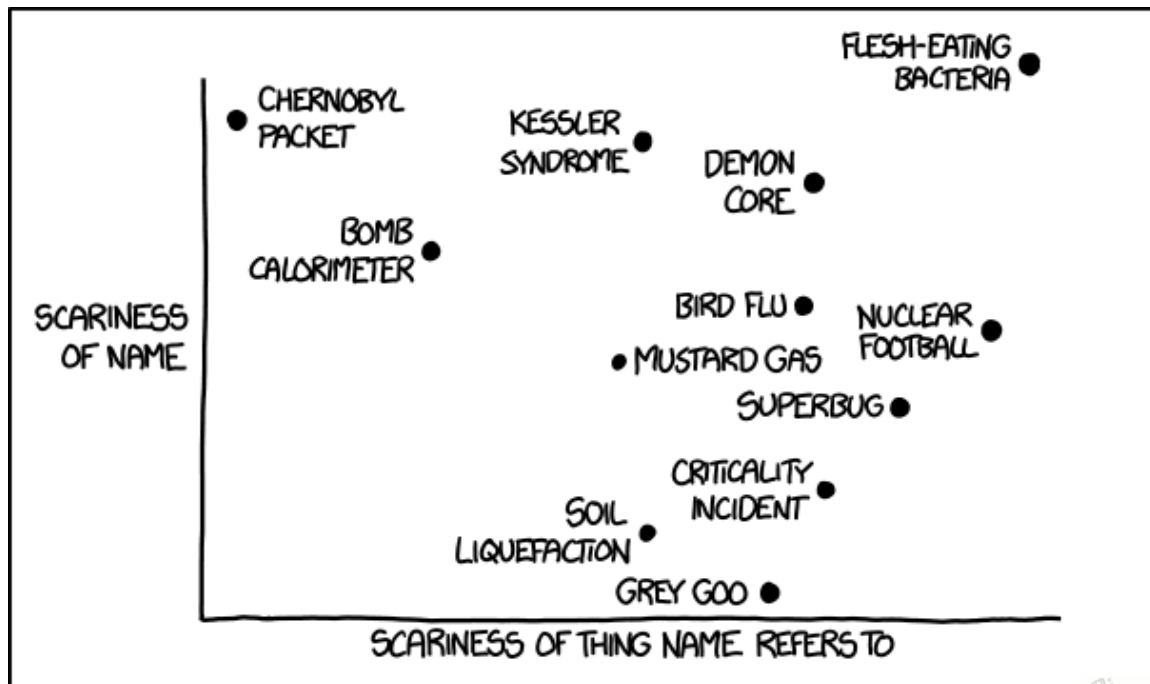
```
SELECT DISTINCT prenom FROM prenoms ;  
SELECT SUM(nombre) FROM prenoms ;  
SELECT COUNT(DISTINCT prenom) FROM prenoms ;  
SELECT annee, SUM(nombre) FROM prenoms GROUP BY annee ;
```

3. Combien de fois votre prénom a-t-il été donné à Paris pendant les années concernées ?
On donnera le résultat trié par années croissantes.
4. Même question avec le prénom des professeurs présents.
5. Donner, pour chaque année, le nombre de prénoms différents qui ont été enregistrés.
6. Quels prénoms ont été donnés exactement 100 fois sur la période complète ?
Sur au moins une année ?
7. Quel est le prénom qui a été le plus donné sur l'ensemble de la période (on donnera aussi le nombre de fois qu'il a été donné) ?
Pour chaque année (pour ce dernier, donner le prénom, l'année et le nombre) ?

⁵Celle-ci est un peu plus dure.

⁶Mais celle-ci n'est pas beaucoup plus compliquée que la précédente.

⁷Là encore, Google est votre ami à condition de lui poser la question gentiment.



Far off to the right of the chart is the Helvetica Scenario.

xkcd.com

Partie V

Notes de colles

La base de données `notes_colles.sqlite` contient trois tables décrivant les colles virtuelles données par des agrégés de la promotion 1930 à des agrégés de la promotion 1950.

1. Écrire le schéma relationnel des différentes tables. Comprendre le lien entre les différentes tables.
2. Déterminer la liste des professeurs.
3. Déterminer celle des élèves.
4. Déterminer le nombre de « 20 » qui ont été attribués, ainsi que le nombre de notes majorant 6.
5. Déterminer les notes de Jacques-Louis Lions (triées selon les semaines croissantes).
6. Refaire la même chose avec cette fois le nom des colleurs associés.
7. Déterminer les quadruplets (élève, prof, note, semaine) pour toutes les colles où la note était supérieure ou égale à 19.
8. Déterminer la moyenne des notes de colle de Jacques-Louis Lions.
9. Parmi tous les élèves, déterminer le nom de ceux ayant eu au moins 10 notes strictement sous la moyenne.
10. Parmi tous les élèves, afficher ceux ayant eu au moins 6 notes strictement supérieures à 18.
11. Déterminer la liste des couples (élève, moyenne).
12. Déterminer le nom du colleur qui tend à donner les meilleurs notes en moyenne.
13. Déterminer pour chaque colleur la moyenne des notes données et la variance (l'écart-type serait plus parlant, mais la fonction `SQRT` n'existe pas en SQLite). Quel est le colleur qui donne le plus souvent les mêmes notes ?
14. Écrire (sans tricher) une requête permettant de calculer la moyenne des moyennes des élèves.

Partie VI

Projet SDSS

Le SDSS (Sloan Digital Sky Survey) est un programme de relevé des objets célestes utilisant un télescope optique dédié de 2,5 mètres de diamètre situé à l'observatoire d'Apache Point, et démarré en 2000. Il a pour but de cartographier 25% du ciel et d'enregistrer les informations relatives à plus de 100 millions d'objets célestes.

Le site web du SDSS (www.sdss.org) permet d'accéder directement à la base de données observationnelle via la page suivante où l'on peut entrer des requêtes SQL

<http://skyserver.sdss3.org/dr10/en/tools/search/sql.aspx>

La structure des tables accessibles via le SDSS est relativement immense comme on peut le voir sur la page suivante qui ne donne que les infos disponibles dans une « super-table⁸ » qui ne concerne que les objets étiquetés « étoile » par le SDSS:

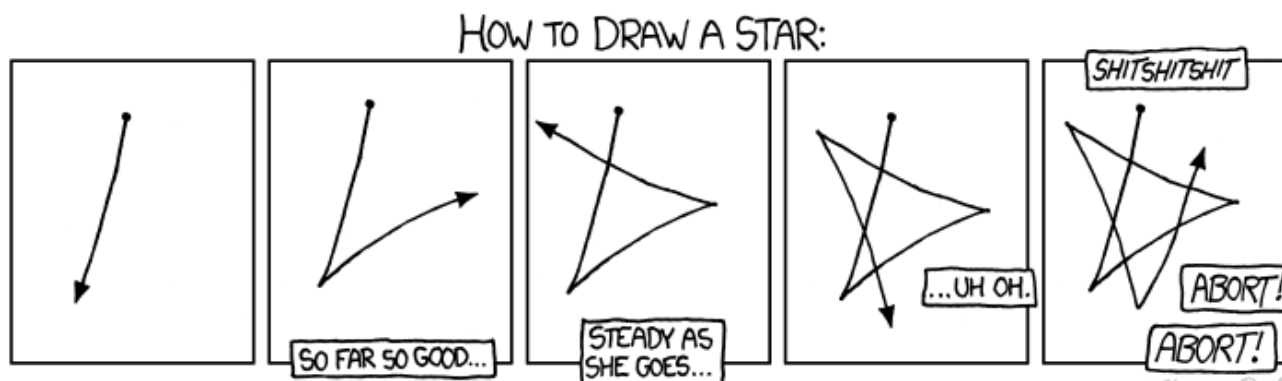
<http://skyserver.sdss3.org/dr10/en/help/browser/browser.aspx#&&history=description+Star+V>

Votre mission, si vous l'acceptez, est de produire le diagramme de Hertzsprung-Russell de la galaxie à partir des données du SDSS. Dans un premier temps, il s'agit de comprendre (sans trop se perdre) comment récupérer les magnitudes en *g* (green) et *r* (red) d'objets identifiés comme étant des étoiles par le SDSS. Ne soyez pas gourmand au début et ne récupérez qu'une vingtaine d'objets au total (utilisez `SELECT TOP 20 ...`) pour tester votre chaîne de traitement car il va falloir enregistrer le résultat dans un fichier `.csv`, puis lire ce fichier et demander à Python de tracer la magnitude en *r* en fonction de *g-r* (NB: peut-être peut-on gentiment demander en SQL que le serveur SDSS fasse le boulot à votre place lors de la récupération des données). Une fois que cela fonctionne, vous pouvez étendre votre demande à un millier d'étoiles pour apprécier le résultat.

Dans un second temps, on va regarder si le résultat change notablement quand on prend des étoiles qui font partie d'un même amas stellaire. Par exemple, pour les deux amas suivants:

Nom	Ra	Déc	Rayon
Pal 3	151.3801	0.072	0.7 arcmin
Pal 5	229.0128	-0.1082	3 arcmin

L'ascension droite (Ra pour Right Ascension) et la déclinaison (Déc) sont deux angles (donc exprimés en degrés) semblables aux coordonnées θ et φ du système de coordonnées sphériques. Ils sont suffisants pour repérer un point de manière univoque sur la sphère du ciel. On rappelle que 1 arcmin vaut $1/60^\circ$ de degré. À défaut de prendre un cercle du rayon *R* indiqué autour des coordonnées du centre de chaque amas, on pourra définir un carré de côté $2R$ pour procéder à la sélection.



Screw these 36-degree angles. I'm converting to Judaism.

xkcd.com

⁸On parle de « Vue »