

Bases de données (2/3)

Les 21, 22 et 28 mai 2014

<http://www.mp933.fr/> - stephane@gonnord.org**Buts du TP**

- Vérifier que tout le monde sait maintenant créer/alimenter une base « à l'aide d'un outil interactif ».
- Pratiquer des requêtes avec fonctions d'agrégation.
- Faire des requêtes mixant agrégations et jointures plus ou moins élaborées.

EXERCICE 1 *As usual*

Créer un dossier associé à ce TP ; copier dans ce dossier les fichiers :

titres_reserves.csv prenomns_paris.db notes_colle.db clubs.csv foot0.db

1 Création et alimentation d'une base de données**EXERCICE 2** *Créons une base à la souris.*

1. Créer (depuis `sqliteman`) une base de données contenant deux tables aux schémas relationnels suivants :

<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px; text-align: center;">classe</div> <div style="border: 1px solid black; padding: 2px;"> <u>idc</u> : entier type : texte prof_maths : entier prof_info : entier </div>	<div style="border: 1px solid black; padding: 2px; margin-bottom: 5px; text-align: center;">prof</div> <div style="border: 1px solid black; padding: 2px;"> <u>idp</u> : entier nom : texte prenom : texte </div>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Réfléchir aux liens entre les tables.

2. Entrer quelques lignes dans chacune de ces tables. Il y aura **entre autres** les tuples :

(841, 'PCSI', 17, 42)
(842, 'PCSI', 23, 42)
(843, 'PCSI', 15, 15)
(17, 'Garcia', 'Antonio')
(23, 'Moynot', 'Olivier')
3. Commiter les changements, quitter `sqliteman`. Relancer `sqliteman` et ouvrir la base précédemment créée ; vérifier que tout le monde est où il faut !

EXERCICE 3 *Importons une table*

1. Ouvrir le fichier `titres_reserves.csv`, qui contient les 1000 titres les plus réservés dans les bibliothèques municipales à Paris en 2013. Observer la nature des informations, et en déduire un schéma relationnel pour inclure ces données dans une base de données.
2. Créer une base de données constituée d'une table répondant à ce schéma.
3. Importer les données du fichier `csv` dans cette table¹. Fermer `sqliteman` puis l'ouvrir, et charger la base de données préalablement créée.
4. Exécuter la requête (en adaptant éventuellement le nom des attributs) :

```
SELECT auteurs,SUM(nombre) AS s
FROM prets
GROUP BY auteurs
ORDER BY s DESC
```

Observer le résultat (dans les 20 premiers titres), et pointer le(s) problème(s) : c'est typique d'une base de données mal conçue/pensée au départ.

1. Oui, vous allez avoir des petites erreurs de signalées ; passons ! Par contre, après l'import, il faut commiter...

2 Prénoms parisiens

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

EXERCICE 4 *Ouvrons les yeux*

Donner le schéma relationnel de cette base (différents attributs des différentes tables).

EXERCICE 5 *De SQL vers le français.*

Pour chacune de requêtes SQL suivantes, donner la traduction « en français », et vérifier la vraisemblance du résultat depuis `sqliteman`.

```
SELECT DISTINCT prenom FROM enregistrement_etat_civil
```

```
SELECT SUM(nombre) FROM enregistrement_etat_civil
```

```
SELECT COUNT(DISTINCT prenom) FROM enregistrement_etat_civil
```

```
SELECT annee,SUM(nombre) FROM enregistrement_etat_civil GROUP BY annee
```

```
SELECT prenom,annee,SUM(nombre) as s
FROM enregistrement_etat_civil
GROUP BY prenom,annee
HAVING s>=300
```

```
SELECT prenom,SUM(nombre) as somme
FROM enregistrement_etat_civil
GROUP BY prenom
HAVING somme>=2000
```

EXERCICE 6 *À vous de jouer.*

- Combien de naissances de filles ont été enregistrées ?
- Combien de fois votre prénom a-t-il été donné à Paris pendant les années concernées ?
On donnera le résultat trié par années croissantes. Attention aux accents, pas toujours bien traités dans la base (le fichier initial était défaillant...).
- Même question avec le prénom du colleur. *Et on ne rigole pas !*
- Donner, pour chaque année, le nombre de prénoms différents qui ont été enregistrés.
- Quels prénoms ont été donnés exactement 100 fois ?
- Quel est le prénom qui a été le plus donné sur l'ensemble de la période ?
- Quel est le prénom *féminin* qui a été le plus donné sur l'ensemble de la période ?

3 Notes de colles

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

EXERCICE 7 *Qu'est-ce qu'on a ?*

Ouvrir cette base. Écrire son schéma relationnel. Comprendre le lien entre les différents attributs des différentes tables.

EXERCICE 8 *C'est parti*

1. Déterminer la liste des noms et prénoms des professeurs... et ceux des élèves.
2. Déterminer le nombre de « 20 » qui ont été attribués, ainsi que les notes majorées par 6.

EXERCICE 9 *Avec jointures*

1. Déterminer les notes de Jacques-Louis Lions (triées selon les semaines croissantes).

2. Refaire la même chose, avec cette fois le nom des colleurs.
3. Déterminer les quadruplets (élève, professeur, note, semaine) pour toutes les colles où la note était supérieure ou égale à 19.

EXERCICE 10 *Des agrégats*

1. Déterminer la moyenne des notes de colle de Jacques-Louis Lions.
2. Déterminer la liste des couples (élève, moyenne).
3. Respirer lentement, puis écrire une requête permettant de calculer la moyenne des moyennes.
Merci de ne pas tricher, et bien calculer une moyenne de moyennes, et non pas la moyenne globale des notes (qui est la même ici, tout le monde ayant eu 25 colles!)
4. Parmi tous les élèves, déterminer le nom de ceux ayant eu au moins 10 notes strictement sous la moyenne.
5. Déterminer les élèves ayant eu une moyenne de 14 ou plus.
6. Parmi tous les élèves, déterminer ceux ayant eu au moins 6 notes strictement supérieures à 18.

4 Des matchs de foot

On veut créer une base de matchs de foot entre clubs européens prestigieux (le Réal, la Juve, Guingamp, etc.) sur quelques dizaines d'années. Disons de 1971 à 2014.

Les impatientes peuvent passer directement au dernier exercice, où on fait des requêtes (absurdes) sur une telle base (absurde)!

EXERCICE 11 *Les clubs*

1. Ouvrir le fichier `clubs.csv`, observer le contenu. Définir sur papier un schéma relationnel permettant de représenter les différents clubs. *On impose un attribut entier qui sera une clé primaire : cet attribut ne prend jamais deux fois la même valeur.*

Pour jeter à la poubelle une éventuelle table, puis en créer une, les commandes SQL sont typiquement :

```
DROP TABLE IF EXISTS clubs
```

```
CREATE TABLE clubs
    (idc integer primary key,
     nom text,
     ...)'''
```

2. À l'aide d'un script Python, créer une base de données, constitué d'une table `clubs`, et l'alimenter à l'aide du fichier `clubs.csv` qu'on lira ligne-à-ligne.

On pourra s'inspirer du squelette fourni au TP précédent : `import sqlite3...`

EXERCICE 12 *Et maintenant, les matchs*

Un match est la donnée de deux équipes *distinctes*, deux scores, une année, un jour dans cette année.

1. Définir sur papier un schéma relationnel pour constituer une table représentant des matchs. Ajouter cette table (vierge, pour le moment) à la base de donnée en cours.

Pour établir le nombre b de buts marqués par une équipe lors d'un match², on tire au hasard $x \in [0, 1]$. Ensuite

- si $x > 3/4$, b vaut 0 ;
- si $5/12 < x \leq 3/4$, b vaut 1 ;
- si $1/12 < x \leq 5/12$, b vaut 2 ;
- sinon, on détermine le plus petit $k \geq 3$ tel que $x \leq \frac{1}{12} \left(1 - \frac{1}{4^{k-2}}\right)$, et b vaut alors k .

2. Écrire une fonction `buts` sans entrée, et retournant un entier déterminé selon les règles précédentes. Vérifier la vraisemblance en faisant 10^6 tirages et en observant les fréquences.

Les probabilités théoriques d'obtenir 0, 1, 2, ... buts sont respectivement $\frac{1}{4}$, $\frac{1}{3}$, $\frac{1}{3}$, $\frac{1}{3 \times 4}$, $\frac{1}{3 \times 4^2}$, etc.

2. on ne se soucie pas du fait de jouer à domicile ou à l'extérieur

3. Remplir la base en réalisant 100 matchs aléatoires pour chaque année entre 1971 et 2014.

La base `foot0.db` est un exemple de telle base générée aléatoirement.

EXERCICE 13 *Et maintenant, jouons un peu !*

1. Que signifie la requête suivante ?

```
SELECT c1.nom, sc1, c2.nom, sc2
FROM clubs c1 JOIN matches JOIN clubs c2
ON c1.idc = eq1 AND c2.idc=eq2
WHERE c1.nom='Real Madrid' OR c2.nom='Real Madrid'
```

2. Déterminer les matchs où Guingamp a mis strictement plus de deux buts à l'extérieur sans gagner.
3. Combien de fois un club français a-t-il marqué au moins 4 buts contre un club italien ?
4. Concevoir quelques requêtes absurdes.

Ça y est : vous êtes prêts pour concevoir des feuilles de TP vous-mêmes !