



BLAISE PASCAL
PT 2021-2022

TD 2 – Informatique

Bases de données

Cette deuxième séance sur les bases de données démarre par une partie de TP sur machine (premier exercice) avant de travailler sur papier sur deux extraits de sujets de concours portant sur les bases de données. Le corrigé sera mis en ligne sur l'ENT *après* la séance.

Exercice 1 : Manipulation de la base de données Mondial

💡 2 | ✂ 2



- ▷ Travail sur machine ;
- ▷ Requêtes simples et moins simples en SQL.

Mondial est une base de données réalisée par l'université de Göttingen, en Allemagne, directement accessible en ligne¹. Elle regroupe pas moins de 43 tables donnant de nombreuses caractéristiques géographiques, culturelles ou encore géopolitiques portant sur 245 états.

1 - Dans un premier temps, on ne travaillera que sur la table **country**. Saisir une requête affichant l'intégralité de cette table. Lister sur votre feuille les différents attributs et leur signification. Lesquels peuvent servir de clé primaire ?

2 - Écrire une requête SQL renvoyant la capitale du Bangladesh.

3 - Écrire des requêtes SQL renvoyant

- 3.a - le nom et la population des pays de plus de cent millions habitants ;
- 3.b - la même liste triée par ordre alphabétique, puis par population décroissante ;
- 3.c - le nombre de ces pays ;
- 3.d - le nombre total d'habitants vivant dans un tel pays ;
- 3.e - la population moyenne de ces pays.

4 - Écrire des requêtes SQL renvoyant

- 4.a - le nom et la densité de population des pays dont ladite densité est inférieure à 10 habitants au km² ;
- 4.b - le nom du pays dont la densité de population est la plus faible au monde.

La table **language** renseigne à propos des langues parlées dans les différents pays. Elle compte trois attributs :

- ▷ l'attribut **country** de la table **language** est une clé extérieure se référant à l'attribut **code** de la table **country** ;
- ▷ l'attribut **name** est le nom de la langue ;
- ▷ l'attribut **percentage** donne la proportion de la population parlant ladite langue.

5 - Écrire une requête SQL renvoyant toutes les langues référencées dont le nom commence par la lettre C.

6 - Écrire des requêtes SQL renvoyant

- 6.a - le nom de chaque pays et le nombre de langues qui y sont parlés ;
- 6.b - la même liste en se restreignant aux pays dans lesquels plusieurs langues sont parlées.

7 - Écrire des requêtes SQL renvoyant

- 7.a - le nombre de pays dans lesquels le français est parlé par une partie au moins de la population ;
- 7.b - le nom de ces pays triés par population ;
- 7.c - le nom de ces pays et le nombre de francophones, triés en fonction de ce nombre. (remarque : certaines données sont manquantes dans la table **language**, ce qui renvoie **null**).

1. <http://www.semwebtech.org/sqlfrontend/> ... mais le langage SQL utilisé présente quelques variantes par rapport à celui que nous utilisons d'habitude.

Exercice 2 : Bilan de santé

écrit PT 2016 | 💡 2 | ✂️ 2



- ▷ Travail sur feuille ;
- ▷ Compréhension de la structure d'une base de données ;
- ▷ Requêtes simples en SQL ;
- ▷ Tracé de figures avec `matplotlib`.

Le sujet porte sur l'étude d'un tensiomètre. La partie modélisation est une partie de SI (ce qui n'est arrivé qu'une seule fois depuis 2015) qui explique comment un tensiomètre procède à ses mesures via une modélisation des jauges de déformation et de la mesure par pont de Wheatstone. La partie d'informatique aborde le traitement des données recueillies, et l'extrait qui nous intéresse traite du stockage des données de différents patients dans une base de données.

L'extrait proposé est assez significatif de ce qui peut être demandé à l'écrit : de la lecture pour comprendre la base de données suivie de quelques requêtes assez simples. Attention à l'écriture de code SQL sur papier : respectez la convention d'écriture des mots clés en majuscule, et n'hésitez pas à revenir à la ligne fréquemment. Indenter le code est possible mais pas obligatoire en SQL.

Les questions manquantes 33 à 36 portaient sur les algorithmes de tri et une étude de complexité, qui feront l'objet de futurs TD.

Le tensiomètre est utilisé dans le domaine hospitalier lors des visites régulières des infirmières dans les chambres des patients. Le tensiomètre doit permettre d'enregistrer les différentes constantes pour chaque patient. Après utilisation, le tensiomètre est connecté au réseau de l'hôpital et synchronise ses mesures avec la base de données des patients.

La base de données est composée de plusieurs tables dont :

- `patients` avec les attributs :
 - `id` : identifiant associé à un patient, clé primaire
 - `numero_secu` : le numéro de sécurité sociale du patient
 - `nom` : le nom du patient
 - `prenom` : le prénom (ou les prénoms) du patient
 - `adresse1` : adresse du patient
 - `adresse2` : complément d'adresse du patient
 - `code_postal` : code postal de la ville du patient
 - `ville` : ville du patient
 - `date_de_naissance` : date de naissance du patient
 - `telephone` : numéro de téléphone
 - ...
- `mesures` avec les attributs :
 - `id` : identifiant associé à une mesure, clé primaire
 - `type` : le type de mesure stockée : résultat d'analyse sanguine, pression artérielle, radio...
 - `datetime` : la date et l'heure de l'examen sous la forme '2016-01-01 00 :00 :00'
 - `pid` : identifiant du patient associé à la mesure
 - `pdias` : pression diastolique en mmHg.
 - `psyst` : pression systolique en mmHg.
 - `pouls` : pouls en pulsations par minute.
 - ...

Remarque : Les tables contiennent bien d'autres attributs, la base de données d'autres tables, seuls les éléments nécessaires à la suite ont été mis en place.

Q29. Rappeler en quelques lignes l'utilité d'une clé primaire dans chacune des tables et dire si un autre attribut des tables aurait pu servir de clé primaire.

Pour le besoin du suivi médical des patients, les médecins doivent pouvoir avoir accès à l'évolution des constantes en fonction du temps lors d'une hospitalisation.

A l'aide d'une interface graphique non étudiée ici, le médecin peut sélectionner les données à extraire entre deux dates qui seront stockées dans les variables `time1` et `time2` stockées dans le format de `datetime`. Il souhaite tracer les évolutions des pressions systolique, diastolique et le pouls entre ces deux dates.

Q30. Donner la requête SQL à envoyer au serveur de gestion de la base de données permettant de récupérer la valeur des champs `datetime`, `pdias`, `psyst` et `pouls` entre les deux dates `time1` et `time2` (stockées dans le format de `datetime`).

La requête précédente renvoie toutes les mesures sans distinction de type de mesures et sans que ce soit lié à un unique patient.

Le médecin a sélectionné un patient avec son identifiant unique `id`, cette valeur est stockée dans la variable `id_patient`.

Q31. Donner la requête SQL à envoyer au serveur de gestion de la base de données permettant de récupérer la valeur des champs précédents entre les deux mêmes dates, pour le patient `id_patient` et pour un type de mesures `'tension'`.

La requête est envoyée et traitée à l'aide d'une fonction `traitement_requete(requete)` qui retourne un tableau `resultat_requete` dont la première colonne est le champ `datetime`, la deuxième le champ `psyst`, la troisième le champ `pdias` et la dernière `pouls`.

On supposera pour simplifier que la valeur de la colonne `datetime` est comprise comme un entier par le langage de programmation retenu et qu'il peut être utilisé directement comme abscisse des graphiques à produire dans la question suivante.

Q32. Donner la séquence d'instructions permettant de tracer un graphique comportant l'évolution des pressions diastolique et systolique ainsi que le pouls en fonction du temps. Vous veillerez à mettre des titres et légendes sur le graphique. Vous utiliserez des styles de lignes différents pour chaque ligne dont vous explicitez en commentaire le résultat visuel (couleur, type de trait : continu, pointillé...). Des éléments de documentation sont donnés en annexe page 11.

Le médecin souhaite obtenir les nom, prénom et le numéro de téléphone des patients qui ont eu une pression systolique supérieure strictement à 160 mmHg, une pression diastolique supérieure strictement à 110 mmHg, avec un pouls compris strictement entre 100 et 150.

Q37. Ecrire la requête SQL à envoyer au serveur de gestion de la base de données permettant de répondre à la demande du médecin.

Annexe - documentation partielle

Plot lines and/or markers to the Axes. `args` is a variable length argument, allowing for multiple `x, y` pairs with an optional format string (`fmt`). For example, each of the following is legal :

```
plot(x, y)           # plot x and y using default line style and color
plot(x, y, 'bo')    # plot x and y using blue circle markers
```

An arbitrary number of `x, y, fmt` groups can be specified, as in :

```
plot(x1, y1, 'g^', x2, y2, 'g-')
```

Return value is a list of lines that were added.

By default, each line is assigned a different style specified by a 'style cycle'.

The following format string characters are accepted to control the line style or marker :

character	description	character	description
'-'	solid line style	'3'	tri_left marker
'_'	dashed line style	'4'	tri_right marker
'-.'	dash-dot line style	's'	square marker
':'	dotted line style	'p'	pentagon marker
''	point marker	'*'	star marker
','	pixel marker	'h'	hexagon1 marker
'o'	circle marker	'H'	hexagon2 marker
'v'	triangle_down marker	'+'	plus marker
'^'	triangle_up marker	'x'	x marker
'<'	triangle_left marker	'D'	diamond marker
'>'	triangle_right marker	'd'	thin_diamond marker
'1'	tri_down marker	' '	vline marker
'2'	tri_up marker	'_'	hline marker

The following color abbreviations are supported :

character	color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

Line styles and colors are combined in a single format string, as in 'bo' for blue circles.

`xlabel(s)` : Set the x axis label of the current axis with the string `s`.

`ylabel(s)` : Set the y axis label of the current axis with the string `s`.

`title(s)` : Set a title of the current axes with the string `s`.

`legend(*args)` : Places a legend on the axes. `args` must be a list of string.

Exercice 3 : Paludisme

écrit Mines MP-PC-PSI 2016 | 💡 2 | ✂ 2



- ▷ Travail sur feuille ;
- ▷ Compréhension de la structure d'une base de données ;
- ▷ Requêtes simples en SQL.

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	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

...

❑ **Q5** – 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.

❑ **Q6** – 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'exprime 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.

❑ **Q7** – Écrire une requête en langage SQL qui détermine le taux d'incidence du paludisme en 2011 pour les différents pays de la table **palu**.

❑ **Q8** – 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).

❑ **Q9** – Écrire une requête en langage SQL donnant le nom de chaque pays et le nombre de décès dus au paludisme en 2010 trié par ordre croissant de nombre de décès.