

Prise en main d'une base de données

1 Lancement du Client SQL

Pour interroger un **serveur** de base de données, il faut utiliser un programme particulier, appelé **client**. Pour ce TP, le client utilisé s'appelle PhpMyAdmin. Il est accessible depuis les ordinateurs des salles informatiques du lycée à partir d'un navigateur web à l'adresse "https://sql.saintlouis.local". Pour vous y connecter, vous aurez besoin d'un identifiant : "user1" et du mot de passe "user1".

Nous utiliserons la base de données **LSTL_world** qui contient les données de la base "World", librement accessible sur internet (<http://dev.mysql.com/doc/world-setup/en/index.html>). Elle comprend trois tables :

- La table **country** contient les pays du monde, identifiés par un code à trois lettres ainsi que de multiples informations sur chaque pays.
 - La table **city** recense quelques grandes villes mondiales, le code du pays auquel elles appartiennent et d'autres informations
 - La table **countryLanguage** contient des informations sur les langues parlées dans chaque pays.
1. Identifier, pour chaque table, les attributs, clés primaires ainsi que les relations entre ces trois tables (clés étrangères).

2 Premières requêtes simples

1. Afficher la liste des pays du monde
2. Afficher la liste des pays du monde ainsi que leur surface
3. Afficher la liste des villes de plus de 300000 habitants
4. Afficher la liste des pays Européens

3 Requêtes utilisant les fonctions d'agrégation

1. Donner le nombre de pays où l'espérance de vie est supérieure à 70 ans.
2. Donner, pour chaque code pays à trois lettres, le nombre de langues parlées dans ce pays.

3. Afficher la liste des régions d'Europe avec, pour chaque région, le nombre de pays qui la compose.
4. Donner, pour chaque code pays à trois lettres, la moyenne du nombre d'habitants des villes de ce pays présentes dans la base.
5. Donner, pour chaque code pays à trois lettres, le pourcentage de la population parlant une langue officielle
6. Afficher pour chaque continent, le PIB le plus important

4 Jointures

1. Afficher la liste des noms des pays avec leur capitale
2. Afficher pour chaque couple (Continent, Région) le nombre de langues différentes parlées dans les pays de cette région.
3. Afficher la liste des noms de pays dont au moins 50% de la population parle une langue officielle.

5 Sous-requêtes

1. Afficher la liste des villes de France
2. Afficher la liste des noms des pays Européens dont la population est inférieure à la moyenne mondiale.
3. Afficher, pour chaque code pays, le nombre de villes dont la population dépasse la population moyenne des villes de France
4. Afficher la liste des noms des pays Européens ainsi que l'écart entre leur PIB par habitant et la moyenne Européenne.
5. Afficher pour chaque continent, le pourcentage de pays pour lesquels le PIB par habitant est supérieur à la moyenne du continent
6. Afficher la liste des villes dont le nom correspond à au moins deux pays. On pourra, dans un second temps afficher simultanément les codes de deux pays correspondant.
7. Afficher, pour chaque région, le pays le plus peuplé.
8. Afficher, pour chaque région, le deuxième pays le plus peuplé.

Centrale 2018

On dispose d'une version plus générale de la fonction `simulation` pour laquelle toutes les particules ne sont plus nécessairement identiques. Cette fonction enregistre ses résultats dans une base de données dont la structure est donnée figure 4.

SIMULATION	
SI_NUM	integer
SI_DEB	datetime
SI_DUR	float
SI_DIM	integer
SI_N	integer
SI_L	float

REBOND	
SI_NUM	integer
RE_NUM	integer
PA_NUM	integer
RE_T	float
RE_DIR	integer
RE_VIT	float
RE_P	float

PARTICULE	
PA_NUM	integer
PA_NOM	varchar(100)
PA_M	float
PA_R	float

Figure 4 Structure physique de la base de données des résultats de simulation

Cette base comporte les trois tables suivantes :

- la table `SIMULATION` donne les caractéristiques de chaque simulation effectuée. Elle contient les colonnes
 - `SI_NUM` numéro d'ordre de la simulation (clef primaire)
 - `SI_DEB` date et heure du lancement du programme de simulation
 - `SI_DUR` durée (en secondes) de la simulation (il ne s'agit pas du temps d'exécution du programme, mais du temps simulé)
 - `SI_DIM` nombre de dimensions de l'espace de simulation
 - `SI_N` nombre de particules pour cette simulation
 - `SI_L` (en mètres) taille du récipient utilisé pour la simulation
- la table `PARTICULE` des types de particules considérées. Elle contient les colonnes
 - `PA_NUM` numéro (entier) identifiant le type de particule (clef primaire)
 - `PA_NOM` nom de ce type de particule
 - `PA_M` masse de la particule (en grammes)
 - `PA_R` rayon (en mètres) de la particule
- la table `REBOND`, de clef primaire (`SI_NUM`, `RE_NUM`), liste les chocs des particules avec les parois du récipient. Elle contient les colonnes
 - `SI_NUM` numéro d'ordre de la simulation ayant généré ce rebond
 - `RE_NUM` numéro d'ordre du rebond au sein de cette simulation
 - `PA_NUM` numéro du type de particule concernée par ce rebond
 - `RE_T` temps de simulation (en secondes) auquel ce rebond est arrivé

- `RE_DIR` paroi concernée : entier non nul de l'intervalle $[-SI_DIM, SI_DIM]$ donnant la direction de la normale à la paroi. Ainsi -2 désigne la paroi située en $y = 0$ alors que 1 désigne la paroi située en $x = L$
- `RE_VIT` norme de la vitesse de la particule qui rebondit (en $m \cdot s^{-1}$)
- `RE_VP` valeur absolue de la composante de la vitesse normale à la paroi (en $m \cdot s$)

1. Ecrire une requête SQL qui donne le nombre de simulations effectuées pour chaque nombre de dimensions de l'espace de simulation.
2. Ecrire une requête SQL qui donne, pour chaque simulation, le nombre de rebonds enregistrés et la vitesse moyenne des particules qui frappent une paroi.
3. Ecrire une requête SQL qui, pour une simulation n donnée, calcule, pour chaque paroi, la variation de quantité de mouvement due aux chocs des particules sur cette paroi tout au long de la simulation. On se rappellera que lors du rebond d'une particule sur une paroi la composante de sa vitesse normale à la paroi est inversée, ce qui correspond à une variation de quantité de mouvement de $2m|v_{\perp}|$ où m désigne la masse de la particule et $v_{\perp}$ la composante de sa vitesse normale à la paroi.

Mines 2018

On dispose d'une base de données relationnelle **Vagues**.

La première table est **Bouee**. On se limite aux attributs suivants : le numéro d'identification **idBouee**, le nom du site **nomSite**, le nom de la mer ou de l'océan **localisation**, le type du capteur **typeCapteur** et la fréquence d'échantillonnage **frequence**.

Bouee				
idBouee	nomSite	localisation	typeCapteur	frequence
831	Porquerolles	Mediterranee	Dw non directionnelle	2.00
291	Les pierres noires	Mer d'iroise	Dw directionnelle	1.28
...	...	...	...	...

La seconde table est **Campagne**. On se limite aux attributs suivants : le numéro d'identification **idCampagne**, le numéro d'identification de la bouée **idBouee**, la date de début **debutCampagne** et la date de fin **finCampagne**.

Campagne			
idCampagne	idBouee	debutCampagne	finCampagne
08301	831	01/01/2010 00h00	15/01/2010 00h00
02911	291	15/10/2005 18h30	18/10/2005 08h00
...	...	...	...

La troisième table est **Tempete**. Les informations fournies relatives à un événement "tempête" sont les suivantes :

- date de début et fin de tempête;
- évolution des paramètres $H_{1/3}$ et H_{max} en fonction du temps
- Le détail de certains paramètre non définis ici, obtenus au pic de tempête.

On se limite aux attributs suivants : le numéro d'identification de la tempête **idTempete**, le numéro d'identification de la bouée **idBouee**, la date de début **debutTempete**, la date de fin **finTempete**, la valeur maximale de hauteur de vague **Hmax**.

Tempete				
idTempete	idBouee	debutTempete	finTempete	Hmax
083010	831	07/01/2010 20h00	09/01/2010 15h30	5.3
029012	291	16/10/2005 08h30	18/10/2005 09h00	8.5
...	...	...	...	...

Le schéma de la base de donnée est donc : **Vagues=Bouee,Campagne,Tempete**.

Formuler les requêtes SQL permettant de répondre aux questions suivantes :

1. "Quels sont le numéro d'identification et le nom de site des bouées localisées en Méditerranée?"
2. "Quel est le numéro d'identification des bouées où il n'y a pas eu de tempêtes?"
3. "Pour chaque site, quelle est la hauteur maximale enregistrée lors d'une tempête?"

X-ENS 2017

On suppose maintenant que l'on représente les points du problème à l'aide d'une base de données. Cette base comporte deux tables. La tables **POINTS** contient trois colonnes :

- **id** (clé primaire) qui est un entier naturel unique représentant le point ;
- **x** qui est un entier naturel représentant son abscisse ;
- **y** qui est un entier naturel représentant son ordonnée.

On suppose qu'il n'existe pas deux points d'identifiants distincts et de mêmes coordonnées.

La relation d'appartenance à un ensemble de points est représentée par la table **MEMBRE** à deux colonnes :

- **idpoint**, un entier naturel qui identifie un point ;
- **idensemble**, un entier naturel qui identifie un ensemble de points.

1. Écrire une requête **SQL** qui renvoie les identifiants des ensembles auxquels appartient le point de coordonnées (a, b) .
2. Écrire une requête **SQL** qui renvoie les coordonnées des points qui appartiennent à l'intersection des ensembles d'identifiants i et j .
3. Écrire une requête **SQL** qui renvoie les identifiants des points appartenant à au moins un des ensembles auxquels appartient le point de coordonnées (a, b) .