

Exercice 1

Chaque objet géométrique est représenté de manière numérique par un maillage. On définit les termes suivants :

Maillage : ensemble des facettes qui constituent la géométrie d'un objet. Un maillage sera représenté par une liste de facettes.

Facette : polygone élémentaire qui constitue une partie de la surface d'un objet. Ici, toutes les facettes seront des triangles. Une facette sera représentée par une liste ordonnée de 3 sommets.

Sommet : point délimitant une facette. Il peut être commun à une ou plusieurs facettes. Tout point sera représenté par son vecteur position de coordonnées (x, y, z) .

La figure 1 représente un exemple de maillage simple (tétraèdre), composé de 4 facettes (notées S1 à S4) et de 4 sommets (notés A à D).



FIGURE 1 – Comparaison des deux structures

On souhaite considérer le modèle d'un bateau à moteur. Ce modèle définit plusieurs maillages élémentaires (coque, bouée, échelle, moteur, etc.). Le fichier contenant ce modèle est une base de données relationnelle. En voici un extrait. Les clés primaires sont soulignées.

<u>id</u>	nom
1	coque
2	bouée
3	échelle
4	moteur
...	...

(a) Table `maillage_bateau`

<u>numero</u>	maillage	s1	s2	s3
1	3	1	2	3
2	3	2	4	3
3	2	3	12	5
...

(b) Table `faces`

<u>id</u>	x	y	z
1	0.0	0.0	0.0
2	1.0	0.0	0.0
3	0.0	1.0	0.0
...

(c) Table `sommets`

1. Proposer une requête SQL permettant de compter le nombre de maillages que contient le modèle du bateau.
2. Proposer une requête SQL permettant de récupérer la liste des numéros des facettes (**numero**) du maillage nommé « gouvernail ».
3. Proposer une requête SQL qui renvoie l'écart maximal entre 2 abscisses dans le maillage nommé « coque ».

Exercice 2

On dispose des tables suivantes :

Colonies
<u>id</u>
nom
lieu
date
capacite
id_animateur1
id_animateur2
id_animateur3
id_animateur4

Animateurs
<u>id</u>
nom
prenom
age
talent

Inscrits
<u>numero</u>
nom
prenom
age
id_colonie
preference

Les attributs se comprennent aisément... La capacité d'une colonie est le nombre d'enfants qu'elle peut accueillir, et ne tient pas compte des inscrits. Les attributs **talent** et **preference** prennent comme valeurs possibles : 'sport', 'musique', 'jeux'. Pour simplifier, il n'y a que 4 animateurs par colonie. Les clés primaires sont soulignées. On suppose qu'il y a au moins un inscrit par colonie.

1. Écrire une requête SQL qui donne la liste des inscrits (nom, prénom) par colonie, par ordre alphabétique des noms des colonies et des enfants.
2. Écrire une requête SQL qui renvoie la moyenne d'âge des enfants inscrits. On demande un alias pour cette moyenne.
3. Écrire une requête SQL qui renvoie la liste des colonies (données par leur identifiant) pour lesquelles la moyenne d'âge des participants est inférieure à 12 ans (strictement).
4. Écrire une requête SQL qui donne les 3 colonies (par leur identifiant et leur nom) où il y a le plus d'enfants qui préfèrent la musique (on ne se préoccupe pas des éventuels ex-aequo).
5. Écrire une requête SQL qui donne l'âge minimal et l'âge maximal des enfants inscrits dans une colonie à Chamonix.
6. Écrire une requête SQL qui donne le nombre d'animateurs par catégorie de talent.
7. Écrire une requête SQL qui liste les colonies (par leur identifiant) où il reste de la place.
8. Écrire une requête SQL qui renvoie les nom-prénom des animateurs qui porteraient le même nom-prénom qu'un des inscrits (homonymes entre animateurs et inscrits, donc), pas forcément inscrits à la même colonie.
9. Écrire une requête SQL qui renvoie la ou les colonies de capacité maximale.
10. Écrire une requête SQL qui renvoie les pourcentages d'animateurs par talent (par exemple : sport 50%, musique 25%, jeux 25%).
11. (*Plus difficile ; ne faire que si vous avez terminé le reste.*) Écrire une requête SQL qui renvoie les noms des éventuels homonymes chez les animateurs.