

Bases de données avancées

Sujet 1. Mines 2023

Un exemple basique et récent : je n'ai pas de source LaTeX, je l'ai mis en annexe

Sujet 2. Centrale 2021

Je n'ai pas de source LaTeX, je l'ai mis en annexe

Sujet 3. Centrale 2020

Je n'ai pas de source LaTeX, je l'ai mis en annexe

Sujet 4. X 2019 - TetriX

On souhaite utiliser une base de données pour stocker les résultats obtenus par une communauté de joueurs. On suppose que l'on dispose d'une base de données comportant les tables JOUEURS(`id_j`, `nom`, `pays`) et PARTIES(`id_p`, `date`, `duree`, `score`, `id_joueur`) où :

`id_j` , de type entier, est la clé primaire de la table JOUEURS,

`nom` est une chaîne de caractères donnant le nom du joueur,

`pays` est une chaîne de caractères donnant le pays du joueur,

`id_p` , de type entier, est la clé primaire de la table PARTIES,

`date` est la date (AAAAMMJJ) de la partie,

`duree` , de type entier, est la durée en secondes de la partie,

`score` , de type entier, est le nombre de points marqués au cours de la partie,

`id_joueur` est un entier qui identifie le joueur de la partie.

Question 16. Étant donné une chaîne de caractères `cc` contenant le nom d'un joueur, écrire une requête SQL qui renvoie la date, la durée et le score de toutes les parties jouées par le joueur `cc`, listées par ordre chronologique (au choix, croissant ou décroissant).

Question 17. Étant donné un entier `s` (le score que vient de réaliser une joueuse nommée Alice), écrire une requête SQL qui renvoie la position qu'aura le score `s` dans le classement des parties par ordre de score (on suppose que la dernière partie d'Alice n'a pas encore été insérée dans la table des parties). En cas d'ex aequo pour le score `s` (une ou plusieurs parties déjà présentes ayant le score `s`), le rang sera le même que s'il n'y avait qu'une seule partie avec le score `s`. Par exemple, la requête renverra 1 (le score d'Alice est "1er") si aucun score n'est meilleur que `s`. Autre exemple, si la base de données contient 6 parties dont les scores sont 87 ; 75 ; 75 ; 63 ; 60 ; 60, alors le rang de `s = 75` sera 2, le rang de `s = 70` sera 4 et le rang de `s = 60` sera 5.

Question 18. Écrire une requête SQL qui renvoie le record de France de Tetris couleur, c'est-à-dire le meilleur score réalisé par un joueur dont le pays est la France.

Question 19. Étant donné une chaîne de caractères *cc* contenant le nom d'un joueur (ayant déjà joué au moins une partie de Tetris couleur), écrire une requête SQL qui renvoie le rang du joueur *cc*, c'est-à-dire sa position dans le classement des joueurs par ordre de leur meilleur score dans une partie de Tetris couleur (on traitera les ex aequo de la même manière qu'à la question 17).

Sujet 5. X MP PC 2016 - réseaux sociaux

Une représentation simplifiée, réduite à deux tables, de la base de données d'un réseau social est donnée dans la figure suivante

<i>INDIVIDUS</i>			<i>LIENS</i>	
<i>id</i>	<i>nom</i>	<i>prenom</i>	<i>id1</i>	<i>id2</i>
1	<i>Potter</i>	<i>Harry</i>	1	2
2	<i>Granger</i>	<i>Hermione</i>	2	1

La table *INDIVIDUS* répertorie les individus et contient les colonnes

- *id* (clé primaire), un entier identifiant chaque individu ;
- *nom*, une chaîne de caractères donnant le nom de famille de l'individu ;
- *prenom*, une chaîne de caractères donnant le prénom de l'individu.

La table *LIENS* répertorie les liens d'amitiés entre individus et contient les colonnes

- *id1*, entier identifiant le premier individu du lien d'amitié ;
- *id2*, entier identifiant le second individu du lien d'amitié.

On supposera par ailleurs que pour tout couple (*x*,*y*) dans la table *LIENS*, le couple (*y*,*x*) est également présent dans la table.

Q16. Ecrire une requête SQL qui renvoie les identifiants des amis de l'individu d'identifiant *x*.

Q17. Ecrire une requête SQL qui renvoie les nom et prénom de chacun des amis de l'individu d'identifiant *x*.

Q18. Ecrire une requête SQL qui renvoie les identifiants des individus qui sont amis avec au moins un ami de l'individu d'identifiant *x*.