

---

## Informatique, composition n° 1

---

Le mercredi 9 novembre 2022

### Exercice 1

On souhaite utiliser une base de données pour stocker les résultats obtenus par une communauté de joueurs au jeu "Tetris couleur". 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.
1. Écrire une requête qui donne les 10 premières lignes de la table JOUEURS.
  2. Écrire une requête qui donne les lignes 5 à 14 de la table JOUEURS.
  3. Écrire une requête qui donne le nom et le pays de tous les joueurs.
  4. 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).
  5. 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.
  6. 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.

### Exercice 2

On dispose d'une base de données concernant les astres et planète de notre galaxie qui contient cinq tables :

- la table *Astre* (*nom\_astre*, *diametre*)
- la table *Planete* (*nom\_planete*, *nom\_astre*, *diametre*, *masse*, *periode\_revolution*)
- la table *Astrophysicien* (*nom*, *prenom*, *pays*)
- La table *Asteroide* (*nom\_asteroide*, *masse*, *nom*, *prenom*)
- La table *Collision* (*nom\_astre*, *nom\_planete*, *nom\_asteroide*, *date*)

Un astre (par exemple le soleil) est identifié par un nom et possède un diamètre (exprimé en km).

Une planète (par exemple la Terre) est identifiée par un nom et le nom de l'astre autour duquel elle tourne. Elle possède un diamètre (exprimé en km), une masse (exprimée en kg) et un temps de révolution autour de son astre (exprimé en jours terrestres).

Un astrophysicien est identifié par son nom, son prénom.

Un astéroïde est identifié par un nom et caractérisé par une masse (exprimée en kg) ainsi que le nom et prénom de l'astrophysicien qui l'a découvert.

Enfin une collision est identifiée par le nom d'une planète, le nom de l'astre autour duquel elle tourne et le nom d'un astéroïde. Elle est caractérisée par une date.

1. Donner une requête SQL :

- a) Permettant de donner le nombre d'astres dans la base de données.
- b) Déterminer les astres dont le diamètre dépasse  $10^6$  km.
- c) Donner les planètes qui tourne autour du soleil et dont la période de révolution dépasse 500 jours terrestres.

2. a) Donner une requête SQL permettant d'afficher les planètes concernées par des collisions.

b) Expliquer la requête suivante :

```
SELECT
    nom_planete
FROM
    Planete
WHERE
    Planete.planete_nom NOT IN (
        SELECT Planete.nom_planete
        FROM Planete
            INNER JOIN collision
                ON Planete.nom_planete = Collision.nom_planete
    );
```

3. Écrire une requête SQL permettant :

- a) Donner le nom, prénom et pays des astrophysiciens qui ont découvert un astéroïde
- b) Donner le nom, prénom et pays des astrophysiciens qui n'ont pas découvert d'astéroïde.

4. Écrire une requête SQL permettant de :

- a) Donner le nombre de planètes par astre.
- b) Donner les astres qui possèdent le maximum de planètes.
- c) Donner le nom, prénom et le nombre d'astéroïde que chaque astrophysicien a découvert.