
Informatique, composition n° 1, une correction

Exercice 1

1. Une requête qui donne les 10 premières lignes de la table JOUEURS est :

```
SELECT
    *
FROM
    JOUEURS
LIMIT 10;
```

2. Une requête qui donne les lignes 5 à 14 de la table JOUEURS.

```
SELECT
    *
FROM
    JOUEURS
LIMIT 10 OFFSET 4;
```

3. Une requête qui donne le nom et le pays de tous les joueurs peut-être :

```
SELECT
    nom,
    pays
FROM
    JOUEURS;
```

4. 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) peut-être :

```
SELECT
    date,
    duree,
    score
FROM
    PARTIES
INNER JOIN
    JOUEURS
        ON id_j = id_joueur
WHERE
    nom = cc
ORDER BY
    date ;
```

5. La requête demandé peut-être :

```
SELECT
    COUNT(*) + 1
FROM
    PARTIES
WHERE
    score > s ;
```

6. 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 peut-être :

```

SELECT
    MAX(score)
FROM
    PARTIES
INNER JOIN
    JOUEURS
    ON id_j = id_joueur
WHERE
    pays = "France";

```

Exercice 2

1. a) Une requête SQL permettant de donner le nombre d'astres dans la base de données.

```

SELECT
    COUNT(*)
FROM
    astre;

```

- b) Une requête SQL permettant de déterminer les astres dont le diamètre dépasse 10^6 km peut-être :

```

SELECT
    nom_astre
FROM
    astre
WHERE
    diametre > 1000000 ;

```

- c) Une requête SQL permettant de donner les planètes qui tourne autour du soleil et dont la période de révolution dépasse 500 jours terrestres peut-être :

```

SELECT
    nom_planete
FROM
    Planete
WHERE
    nom_Astre = "Soleil"
    AND periode_revolution > 500;

```

2. a) Une requête SQL permettant d'afficher les planètes concernées par des collisions peut-être :

```

ELECT
    nom_planete
FROM
    Planete
INNER JOIN
    Collision
    ON Planete.nom_planete = collision.nom_planete

```

- b) La requête suivante donne les planètes sans collision :

```

SELECT
    planete.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. a) Une requête SQL permettant de donner le nom, prénom et pays des astrophysiciens qui ont découvert un astéroïde au moins est :

```

SELECT
    astrophysicien.nom,
    astrophysicien.prenom,
    astrophysicien.pays
FROM
    asteroide
INNER JOIN
    astrophysicien
    ON asteroide.nom = astrophysicien.nom ;

```

- b) Une requête SQL permettant de donner le nom, prénom et pays des astrophysiciens qui n'ont pas découvert d'astéroïde peut-être :

```

SELECT
    astrophysicien.nom,
    astrophysicien.prenom
FROM
    astrophysicien
WHERE
    astrophysicien.nom NOT IN (
        SELECT
            astrophysicien.nom
        FROM
            astrophysicien
        INNER JOIN
            astéroïde
            ON asteroide.nom = astrophysicien.nom
    );

```

4. a) Pour obtenir le nombre de planètes par astre on peut faire :

```

SELECT
    nom_astre,
    COUNT(*)
FROM
    planete
GROUP BY
    planete.nom_astre ;

```

- b) Pour obtenir les astres qui possèdent le maximum de planètes on peut faire :

```

SELECT
    nom_astre,
    COUNT(*) AS nbre_planete
FROM
    planete
GROUP BY
    planete.nom_astre
HAVING
    nbre_planete = MAX(nbre_planete);

```

- c) Une requête SQL permettant de donner le nom, prénom et le nombre d'astéroïde que chaque astrophysicien a découvert peut-être :

```
SELECT
    nom,
    prenom,
    COUNT(asteroide.nom) AS nbre_planete
FROM
    astrophysicien
INNER JOIN
    asteroide
        ON asteroide.nom = astrophysicien.nom
GROUP BY
    astrophysicien.nom,
    astrophysicien.prenom ;
```