

Informatique TSI2

Devoir surveillé 01

Mercredi 22 Janvier 2025. Durée 1 heure

SOLUTION

On considère la base de données **TENNIS** munie des deux tables suivantes.

La table **joueurs** contient les attributs :

- **id_joueur** de type entier qui identifie chaque joueur, c'est une clé primaire ;
- **nom** de type chaîne de caractères qui désigne chaque nom de joueur ;
- **annee** de type entier qui désigne l'année de naissance ;
- **nationalite** de type chaîne de caractères qui désigne la nationalité du joueur.

id_joueur	nom	annee	nationalite
1	MURRAY	1987	britannique
2	GULBIS	1988	letton
3	FERDERER	1981	suisse
4	DJOKOVIC	1987	serbe
5	BERDYCH	1985	tchèque
6	NADAL	1986	espagnole
7	THIEM	1993	autrichienne
8	NISHIKORI	1989	japonaise
9	TSONGA	1985	française
10	WAWRINKA	1985	suisse
11	MONFILS	1986	française
12	SIMON	1984	française

La table **tournois** contient les attributs :

- **id_tournoi** de type entier qui identifie chaque tournoi, c'est une clé primaire ;
- **nom** de type chaîne de caractères qui désigne chaque nom de tournoi ;
- **annee** de type entier qui désigne l'année du tournoi ;
- **num_final1** de type entier qui identifie le vainqueur du tournoi, c'est une clé étrangère.
- **num_final2** de type entier qui identifie le perdant de la finale du tournoi, c'est une clé étrangère.
- **num_joueur3** de type entier qui identifie le troisième joueur en demi-finale, c'est une clé étrangère.
- **num_joueur4** de type entier qui identifie le quatrième joueur en demi-finale, c'est une clé étrangère.
- **gain** de type entier qui désigne la somme gagnée par le vainqueur.

id_tournoi	nom	annee	num_final1	num_final2	num_joueur3	num_joueur4	gain
1	ROLAND-GARROS	2016	4	1	7	10	2 000 000
2	ROLAND-GARROS	2015	10	4	9	1	1 800 000
3	ROLAND-GARROS	2015	6	4	1	2	1 650 000
4	OPEN AUSTRALIE	2016	4	8	9	12	2 390 000
5	OPEN AUSTRALIE	2015	4	1	5	10	2 100 000
6	OPEN AUSTRALIE	2014	10	6	3	5	2 050 000

Dans les huit premières questions, donc de **Q1** à **Q8**, on ne s'intéresse qu'à la table **joueurs** et de **Q10** à **Q18**, on s'intéressera aux deux tables et donc on utilisera de **JOIN** quand c'est nécessaire. Par ailleurs, on suppose que l'on ne peut utiliser que la base de données **TENNIS** et ainsi seuls les joueurs, les tournois, les années etc. indiqués dans cette base de données sont connus.

Q1. Écrire une requête SQL qui renvoie toutes les informations de tous les joueurs.

```
SELECT * FROM joueurs ;
```

Q2. Écrire une requête SQL qui renvoie le nom de tous les joueurs français.

```
SELECT joueurs.nom FROM joueurs WHERE nationalite = "française" ;
```

Q3. Écrire une requête SQL qui renvoie la liste des nationalités de tous les joueurs.

```
SELECT DISTINCT nationalite FROM joueurs ;
```

Q4. Écrire une requête SQL qui renvoie la moyenne des années de naissance des joueurs français.

```
SELECT AVG(annee) FROM joueurs WHERE nationalite = "française" ;
```

Q5. Écrire une requête SQL qui renvoie la nationalité, l'année de naissance et le nom des joueurs dont l'année de naissance est supérieure ou égale à 1988.

On renommera dans le résultat la première colonne **Nationalités** >= 1988 et la troisième colonne **Noms** >= 1988 et on triera selon le nom des joueurs de façon croissante.

```
SELECT nationalite AS "Nationalités>=1988", annee, nom AS "Noms>=1988"  
FROM joueurs WHERE annee >= 1988 ORDER BY nom ASC ;
```

Q6. Écrire une requête SQL qui renvoie l'année de naissance et le nom dont l'année de naissance est strictement supérieure à 1980 et de nationalité suisse.

```
SELECT annee, nom FROM joueurs WHERE annee > 1980 AND nationalite = "suisse" ;
```

Q7. Écrire une requête SQL qui renvoie les années de naissance et le nombre de joueurs ayant cette année de naissance.

Les années sont affichées par ordre croissant.

```
SELECT annee, COUNT(*) FROM joueurs GROUP BY annee ORDER BY annee ;
```

Q8. Écrire une requête SQL qui renvoie les identifiants et les noms des 3 premiers joueurs sans le tout premier de la table.

```
SELECT id_joueur, nom FROM joueurs LIMIT 3 OFFSET 1 ;
```

Q9. Qu'appelle t-on une clé primaire ? Qu'appelle t-on une clé étrangère ?

Une clé primaire sert à identifier une ligne de manière unique. Chaque joueur par exemple est identifié par **id_joueur**

Une clé étrangère permet de lier des relations ou tables entre elles. Ainsi la clé **num_final1** permet d'avoir le numéro d'identifiant du joueur qui a gagné la finale du tournoi.

Q10. Écrire une requête SQL qui renvoie le nom du vainqueur de Roland-Garros en 2016.

```
SELECT joueurs.nom FROM joueurs JOIN tournois ON num_final1 = id_joueur  
WHERE tournois.nom = "ROLAND-GARROS" AND tournois.annee = 2016 ;
```

Q11. Écrire une requête SQL qui renvoie pour chaque joueur l'année et le nom des tournois gagnés par celui-ci. Les noms de tous les joueurs sont triés par ordre croissant.

```
SELECT joueurs.nom, tournois.nom, tournois.annee FROM joueurs JOIN tournois  
ON num_final1 = id_joueur ORDER BY joueurs.nom ASC ;
```

Q12. Écrire une requête SQL qui renvoie le nombre de victoires pour chaque joueur.

```
SELECT joueurs.nom, COUNT(*) FROM joueurs JOIN tournois  
ON num_final1 = id_joueur GROUP BY joueurs.nom ;
```

Q13. Écrire une requête SQL qui renvoie le total des gains pour chaque joueur.

```
SELECT joueurs.nom, SUM(tournois.gain) FROM joueurs JOIN tournois  
ON num_final1 = id_joueur GROUP BY joueurs.nom ;
```

Q14. Écrire une requête SQL qui renvoie le gain moyen de chaque joueur d'origine serbe.

```
SELECT joueurs.nom, AVG(tournois.gain) FROM joueurs JOIN tournois  
ON num_final1 = id_joueur WHERE nationalite = "serbe" GROUP BY joueurs.nom ;
```

Q15. Écrire une requête SQL qui renvoie le gain moyen de chaque tournoi.

Les noms des tournois sont rangés par ordre croissant.

```
SELECT tournois.nom, AVG(tournois.gain) FROM tournois  
GROUP BY tournois.nom ORDER BY tournois.nom ;
```

Q16. Écrire une requête SQL qui renvoie la liste des joueurs ayant gagné au moins deux tournois et nés après 1986. On affichera également le nombre de tournois gagnés.

```
SELECT joueurs.nom, COUNT(joueurs.nom) FROM joueurs JOIN tournois  
ON num_final1 = id_joueur WHERE joueurs.annee > 1986 GROUP BY joueurs.nom  
HAVING COUNT(joueurs.nom) >= 2 ;
```

Q17. Écrire une requête SQL qui renvoie pour chaque joueur le nombre de participations à une demi-finale d'un tournoi. On renommera les tables **tournois** et **joueurs** respectivement **j** et **t**.

Les joueurs sont affichés par ordre croissant.

```
SELECT j.nom, COUNT(j.nom) FROM joueurs AS j JOIN tournois AS t  
ON num_final1 = id_joueur OR num_final2 = id_joueur OR num_joueur3 = id_joueur  
OR num_joueur4 = id_joueur GROUP BY j.nom ORDER BY j.nom ;
```