

## TD10 : COMMANDES SQL

(CCINP-PSI-INFO-2024)

*Fichier de la base : « 10. TD\_CCINP\_PSI\_INFO2024.db »*

Un algorithme à base d'IA a été créé pour jouer au jeu de l'awalé (jeu de stratégie très ancien qui fait partie des jeux de semailles).

Afin d'améliorer l'efficacité de l'IA, il est possible pour les premiers coups d'effectuer une recherche dans une base de données relationnelle. En effet, l'exploration en profondeur de l'ensemble des coups possibles est très coûteuse et on préfère s'appuyer sur l'historique des parties pour déterminer les configurations du plateau qui seront les plus avantageuses. La base de données contient des informations sur chaque joueur, ainsi que les parties effectuées entre les joueurs.

Joueur				
<b>id_Joueur</b>	<b>nom</b>	<b>prenom</b>	<b>niveau</b>	<b>naissance</b>
18571	Martin	Jean	2048	23/02/1958
18572	Dupond	Marie	2103	03/01/1972
18573	Develion	Théo	1857	05/10/2004

Partie					
<b>id_Partie</b>	<b>id_joueur1</b>	<b>id_joueur2</b>	<b>resultat</b>	<b>jour</b>	<b>jeu</b>
1	1547	1568	0.5	08/01/2001	'egai...'
2	1204	3	0	12/07/1998	'egaj...'
3	4	2	1	15/07/2018	'egbi...'

La table Joueur contient les attributs suivants :

- **id\_Joueur** : identifiant d'un joueur (entier, clé primaire) ;
- nom : nom du joueur (chaîne de caractères) ;
- prenom : prénom du joueur (chaîne de caractères) ;
- niveau : niveau maximal atteint par le joueur au cours de sa carrière (entier) ;
- naissance : date de naissance du joueur (date).

La table Partie contient les attributs suivants :

- **id\_Partie** : identifiant de la partie (entier, clé primaire) ;
- id\_joueur1 : identifiant du joueur débutant la partie (entier) ;
- id\_joueur2 : identifiant du second joueur (entier) ;
- resultat : 1 est une victoire du joueur1, 0.5 une égalité et 0 une victoire du joueur2 (flottant) ;
- jour : date du jour de la partie (date) ;
- jeu : liste des coups successifs de la partie, sans inversion du plateau, stockée sous forme d'une chaîne de caractères. 'egai...' signifie que le joueur1 a joué la 5e case (d'indice 4) représentée par la lettre 'e', puis le joueur2 a joué la 7e case (d'indice 6) représentée par la lettre 'g', puis le joueur1 a joué la 1re case et le joueur2 a répondu par la 9e case et ainsi de suite.

*Remarque : l'opérateur LIKE est utilisé pour comparer des chaînes de caractères dans la clause WHERE des requêtes SQL. Ce mot-clé permet d'effectuer une recherche sur un modèle particulier. Il est par exemple possible de rechercher les enregistrements dont la valeur d'une colonne commence par telle ou telle lettre. Le caractère \_ (underscore) représente n'importe quel caractère, mais un seul caractère uniquement alors que le caractère pourcentage % peut être remplacé par un nombre quelconque (et possiblement nul) de caractères.*

*Par exemple parmi une recherche dans les communes de France, nom LIKE '\_ff%f%' ne renvoie que Offendorf alors que remplacer le \_ par un % renvoie Pfaffenhoffen et Staffelfelden en plus de Offendorf.*

1. Écrire une requête SQL permettant d'extraire les identifiants des joueurs ayant un niveau strictement supérieur au score 1900.
2. Écrire une requête SQL permettant de déterminer le pourcentage de victoires du joueur1 pour les parties où la case d'indice 0 a été jouée en premier.
3. Écrire une requête SQL permettant d'afficher le nom et le prénom des 3 joueurs ayant le niveau le plus élevé.
4. Écrire une requête SQL permettant de déterminer les joueurs ayant plus de cent victoires lorsqu'ils commencent la partie. La requête doit renvoyer le nom, le prénom et le nombre de victoires de ces joueurs classés par ordre décroissant du nombre de victoires.

---

## CORRECTION

---

1. 

```
SELECT id_Joueur FROM Joueur WHERE niveau > 1900
```
2. 

```
SELECT COUNT(resultat)*100
/ (SELECT COUNT(resultat) FROM Partie WHERE jeu LIKE "a%")
AS Pourcentage FROM Partie WHERE resultat = 1 AND jeu LIKE "a%" ;
```
3. 

```
SELECT nom, prenom FROM Joueur ORDER BY niveau DESC LIMIT 3 ;
```
4. 

```
SELECT j.nom, j.prenom, COUNT(p.resultat) FROM Joueur AS j JOIN Partie AS p
ON j.id_Joueur=p.id_joueur1 WHERE p.resultat=1
GROUP BY j.id_Joueur
HAVING COUNT(p.resultat)>100 ORDER BY COUNT (p.resultat)>100 DESC
```