



Un quotidien régional a demandé à la SDSM (Secure Data Systems Modeling) de concevoir une base de données pour stocker les résultats des matchs de basket du championnat des équipes locales pour la saison en cours. Voici le modèle relationnel fourni par votre chef de projet qui vous demande de vous l'approprier en répondant à quelques questions et en rédigeant quelques requêtes SQL :

Club (id, nom, nomContact, telephoneContact, nbGagnes, nbNuls, nbPerdus)

Clé primaire: id

Joueur (id, nom, prenom, dateNaissance, idClub)

Clé primaire : id

Clé étrangère : idClub en référence à id de Club

Matches (id, dateM, heure, idClubVisite, scoreClubVisite, idClubVisiteur, scoreClubVisiteur)

Clé primaire : id

Clé étrangère : idClubVisite en référence à id de Club

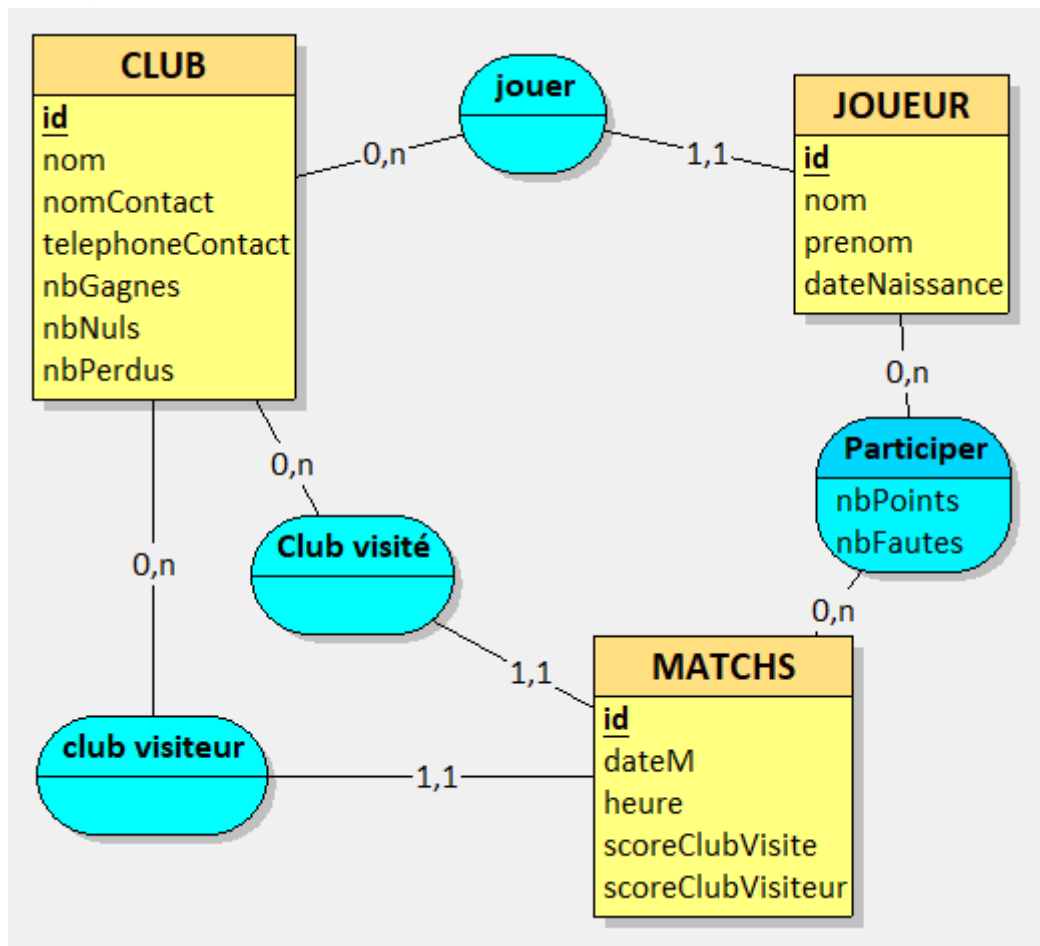
Clé étrangère : idClubVisiteur en référence à id de Club

Participer (idMatch, idJoueur, nbPoint, nbFaute)

Clé primaire : idMatch, idJoueur

Clé étrangère : idMatch en référence à id de Match

Clé étrangère : idJoueur en référence à id de Joueur



Partie 1 : Analyse du schéma de données

1) Quelle est la clé primaire de la table Club ? Que devra vérifier le SGBD ?

La clé primaire de la table Club est représenté par id, le SGBD vérifie qu'il n y a pas 2 fois le meme identifiant

2) Quelle est la clé primaire de la table Participer ? Que devra vérifier le SGBD ? Un joueur pourrait-il, par erreur, être inscrit 2 fois au même match ?

La clé primaire, est composée des 2 attributs, id match, et id joueur. Le SGBD vérifie que le couple ne se répète pas 2 fois.

1) Expliquer en quoi **idClub** dans la table Joueur est un attribut particulier et quel est son rôle ? Que devra vérifier le SGBD ? Comment appelle-t-on les contraintes associées ce concept ?

C'est une clé étrangère, le SGBD vérifie si le club existe dans la base de données ou non.

3) En utilisant deux phrases, décrire le lien entre les tables Joueur et Club.

Il faut expliquer dans les 2 sens, un joueur joue dans un seul club, et un club peut avoir plusieurs joueurs.

4) Un joueur peut-il ne participer à aucun match ?

Joueur peut ne participer à aucun match, car il y a la cardinalité 0,n

5) Un joueur donne sa démission, est-il possible de le supprimer de la table Joueur ?

Non, certain SGBD proposent la suppression en cascade (càd supprimer tous les enregistrements étants associés à cette clé étrangère, mais la majorité ne permettent pas de supprimer)

Partie 2 : Requêtes SQL – sans implémenter la base de données.

1) Afficher la date, l'heure, le n° du club visité, le n° du club visiteur du match n°2.

```
SELECT id, dateM, idClubVisite, idClubVisteur
FROM matchs
WHERE id=2;
```

2) Afficher la liste des dates du championnat par ordre chronologique.

```
SELECT dateM
FROM Matches
ORDER BY dateM ;
```

3) Quels seront les matchs du club n°1 en janvier 2024 ? Plusieurs solutions selon le sgbd utilisé.

```
SELECT id, idClubVisite, idClubVisiteur
FROM Matches
WHERE idClubVisite = 1 OR idClubVisiteur = 1
AND dateM LIKE '%%/01/2024'
AND dates BETWEEN '01/01/2024' AND '31/01/2024'
AND YEAR(dateM) = 2024 AND MONTH(dateM) = 1
```

Pour le match n°1, afficher le n° et le nombre de points des joueurs, le tout trié par nombre de points décroissant.

4) Afficher le nombre total de fautes effectuées lors du match n°1.

```
SELECT idJoueur, nbpoint
FROM Participer
WHERE idmatch = 1
ORDER BY nbpoint DESC
```

5) Afficher le nombre de joueurs licenciés dans le club n°3.

```
SELECT SUM(nbFaute)
FROM Participer
WHERE IdMatch = 1 ;
```

6) Pour chaque joueur (n°), afficher le total des points marqués.

```
SELECT Count(*)
FROM Joueur
WHERE idClub = 3 ;
```

7) Pour chaque club (n°) quelle est la moyenne des scores à l'extérieur (club visiteur), à trier par moyenne décroissante.

```
SELECT idJoueur, SUM(nbPoint)
FROM Participer
GROUP BY idJoueur ;
```

8) On veut joindre le contact du club visité du match n°6, quel est son nom et son téléphone ?

```
SELECT idClubVisiteur, AVG(scoreClubVisiteur)
FROM Match
GROUP BY idClubVisiteur
ORDER BY 2 DESC
```

9) Pour chaque joueur (nom, prénom, nom du club), afficher le nombre de points et le nombre de fautes, à trier par nombre de points décroissant.

```
SELECT nomContact, telephoneContact
FROM Club
JOIN Matches ON idClubVisite = Club.id
WHERE Match.id = 6;
```

10) Lksdfjsd

```
SELECT Joueur.nom, prenom, Clubs.nom, SUM(nbPoints), SUM(nbFaute)
FROM Joueur
JOIN Participer ON Joueur.id = Participer.idJoueur
JOIN Club ON Joueur.idClub = Club.id
GROUP BY Joueur.nom, prenom, Clubs.nom
ORDER BY 4 DESC
```

Partie 3 : Implantation de la base de données

Le chef de projet vous explique qu'il a choisi d'implanter la base de données en SQLite (pour une future application mobile). Il vous demande de créer une nouvelle base de données "**Basket.db**" avec l'outil DB Browser et d'exécuter le script qu'il vous a fourni "**Script Basket sqlite.sql**".

Examinez le résultat obtenu et indiquez dans quel état se trouve la base de données.

Partie 4 : Test de requêtes SQL

Testez les requêtes 1 – 2 – 3 – 6 – 9

Pour information en SQLite une date est de format TEXT mais formatée de la façon suivante : 'YYYY-MM-DD' par exemple '2024-01-13' pour le 13 janvier 2024. Une date en SQL est entourée de quotes (ou apostrophes)

Partie 5 : Insertion de données – modification de données

A partir de DB Browser insérer des données **plausibles et cohérentes** pour chaque match (scores, participations, nombres de points, de fautes et mise à jour des matchs gagnés, nuls ou perdus).

Partie 6 : Test de requêtes SQL

Testez les requêtes 4 – 5 – 7 – 8 – 10

Partie 7 : Nouvelle requête SQL

Si l'on considère qu'un match gagné remporte 3 points, un match nul 1 point, établir une requête SQL qui affiche le classement des clubs en fin de saison. La forme du résultat attendu est le suivant :

	nom	Total	Gagnés	Nuls	Perdus
1	Niederhausbergen	12	4	0	0
2	Mittelhausbergen	4	1	1	2
3	Oberhausbergen	1	0	1	3

Votre résultat sera différent mais devra être conforme avec les données insérées en fin de saison...