



DEUXIÈME ANNÉE INFORMATIQUE

PROJET SEMESTRE 7 - SYSTÈMES DE GESTION DE BASES DE DONNÉES

Fédération Sportive de Hand-Ball

Auteurs:

Chaymae AMMI
Enzo MEZZASALMA
Mamadou MOUSSA GUIMBA
Walid LAAZIRI

Encadrants:

Sylvain LOMBARDY
Mohamed MOSBAH

Table des matières

1	Introduction	2
2	Réalisation du modèle conceptuel	2
2.1	Les entités mises en jeu	2
2.2	Le modelé entité-association	2
2.3	Le modèle conceptuel	3
3	Réalisation du modèle relationnel	4
3.1	Du conceptuel au relationnel	4
3.2	Modèle Relationnel final	5
4	Implémentation de la base de données	6
4.1	Choix du SGBD	6
4.2	Requêtes et fonctionnalités	6
5	Interface graphique	8
6	Conclusion	11

1 Introduction

Afin de mettre en pratique les notions acquises en module de Systèmes de Gestion de Bases de Données, nous avons été amenés à faire un projet de groupe qui consiste à réaliser une base de données qui gère les matchs d'une fédération sportive de Hand-Ball, tout ce avec le langage SQL.

Le projet s'est fait en plusieurs étapes en commençant par la modélisation des données où nous avons réalisé les schémas conceptuel et relationnel, passant par l'implémentation de la base de données et arrivant à la création d'une interface Web qui offre des fonctionnalités de manipulation simple des données.

Toutes ces étapes vont être expliquées et détaillées, nos différents choix justifiés et l'interface présentée dans ce rapport.

2 Réalisation du modèle conceptuel

Dans cette première partie nous allons présenter notre modèle conceptuel accompagné des justifications de nos différents choix.

2.1 Les entités mises en jeu

Les rencontres sportives se font entre différents clubs de la fédération sportive. Chaque club possède un bureau et comprend plusieurs équipes de différentes catégories et chacune d'elles possède un ou plusieurs entraîneurs. Les rencontres sont faites à une date donnée et se déroulent entre deux équipes. Par conséquent les entités qui définissent le cadre de notre base sont les suivantes : le personnel, les joueurs, les clubs, les entraîneurs, les équipes, les catégories et les rencontres.

2.2 Le modelé entité-association

Les associations trouvées pour le modèle conceptuel sont les suivantes:

- **Accompagner**: Toute personne de l'entité Personnel peut accompagner au moins un ou plusieurs clubs. Inversement, tout club peut être accompagné par au moins un personnel. Il est donc question d'une association (1,N) dans les deux sens.
- **Entrer**: Un joueur peut intégrer un et uniquement un seul club, il s'agit d'une association (1,1). Dans l'autre sens, un club peut avoir au moins un ou plusieurs joueurs, c'est une association (1,N).
- **Entrer au club**: Un club peut posséder un à plusieurs entraîneurs, c'est une association (1,N). Un entraîneur ne peut exercer sa fonction que dans un club, il est donc question d'une association (1,1).
- **Posséder**: Chaque club possède au moins une ou plusieurs équipes, c'est une association (1,N). Inversement, chaque équipe appartient à un et un seul club, il s'agit d'une association (1,1).
- **Jouer**: Tout joueur peut jouer dans une ou plusieurs rencontres et inversement dans chaque rencontre il y a au moins un joueur. Il s'agit d'une association (1,N) dans les deux sens.
- **Faire partie**: Chaque joueur peut faire partie d'au moins une équipe et dans chaque équipe il y a au moins un joueur. C'est une association (1,N) dans les deux sens.
- **Entraîner**: Un entraîneur peut entraîner au moins une équipe et toute équipe peut avoir au moins un entraîneur. Dans les deux sens l'association est de cardinalité (1,N).
- **Participer**: Une équipe peut participer à aucune ou plusieurs rencontres, c'est une association (0,N). Toute rencontre se déroule entre deux équipes, il est donc question d'une association (1,N).

- **Appartenir**: Une équipe appartient à une et une seule catégorie, c'est une association (1,1).
Et nous pouvons avoir plusieurs équipe pour une catégorie , il s'agit d'une association (1,N).

2.3 Le modèle conceptuel

La conception des entités-associations détaillées dans la sous-partie précédente nous a permis de réaliser le schéma conceptuel qui suit:

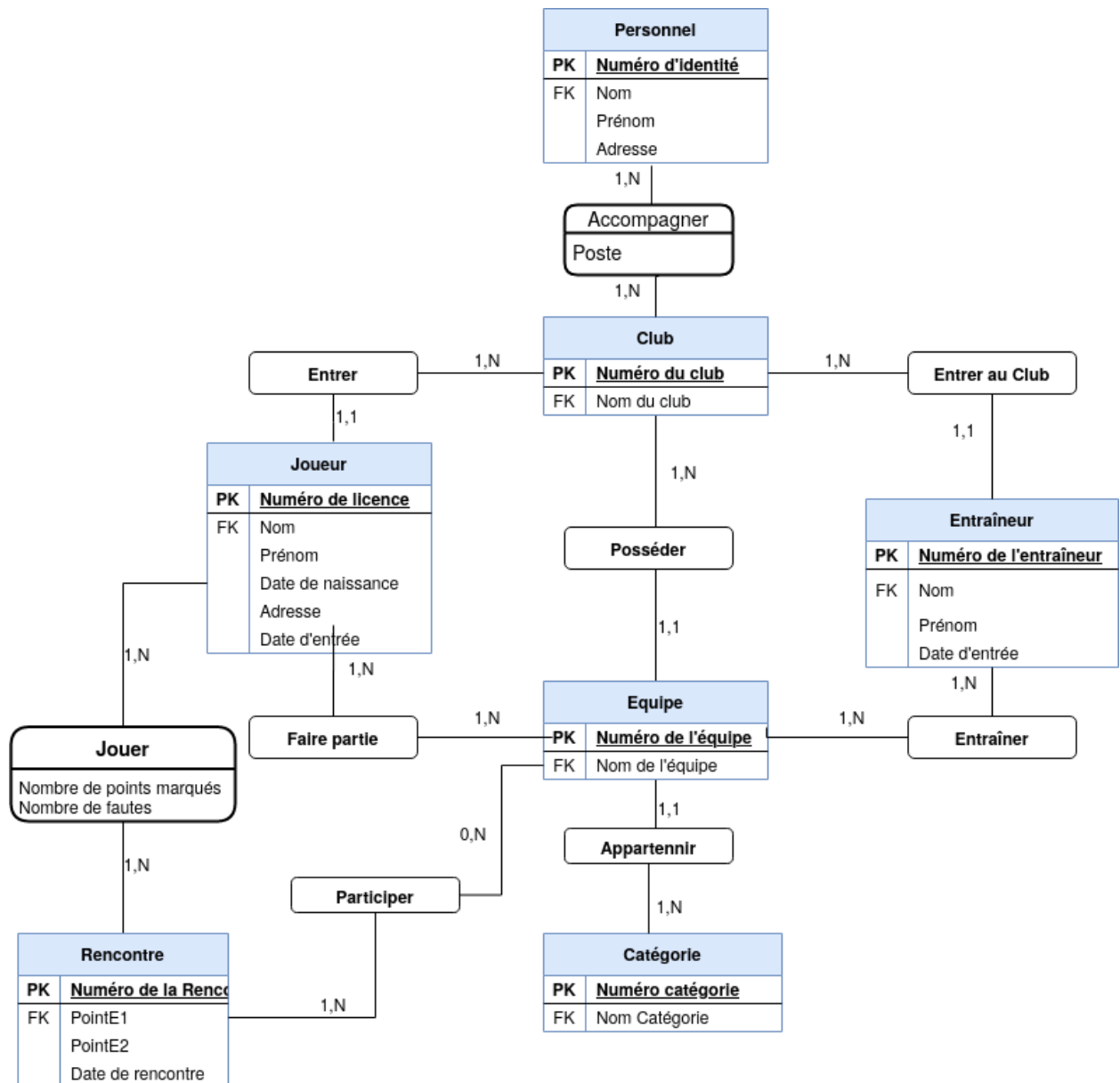


Figure 1 – Modèle Conceptuel Final

3 Réalisation du modèle relationnel

Après avoir présenté en détail notre modèle conceptuel de données, nous allons expliquer le passage de celui-ci au modèle relationnel de données tout en justifiant les modifications apportés.

3.1 Du conceptuel au relationnel

Pour traduire le modèle conceptuel en modèle relationnel nous avons respecté les règles suivantes:

Règle Entité-Table

Toute entité devient une table, les attributs ses colonnes et l'identifiant sa clé primaire:

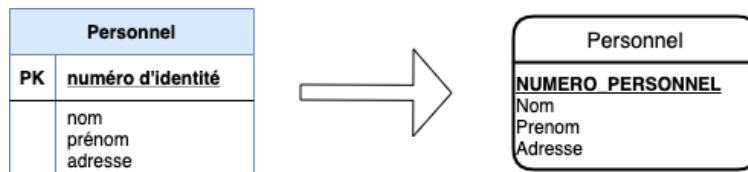


Figure 2 – Entité-Table

Règle Association de type (1,N)

Une association binaire de type (1,N) disparaît au profit d'une clé étrangère dans la table côté (0,1) ou (1,1) qui référence la clé primaire de l'autre table. Cette clé étrangère ne peut pas recevoir la valeur vide si la cardinalité est (1,1):

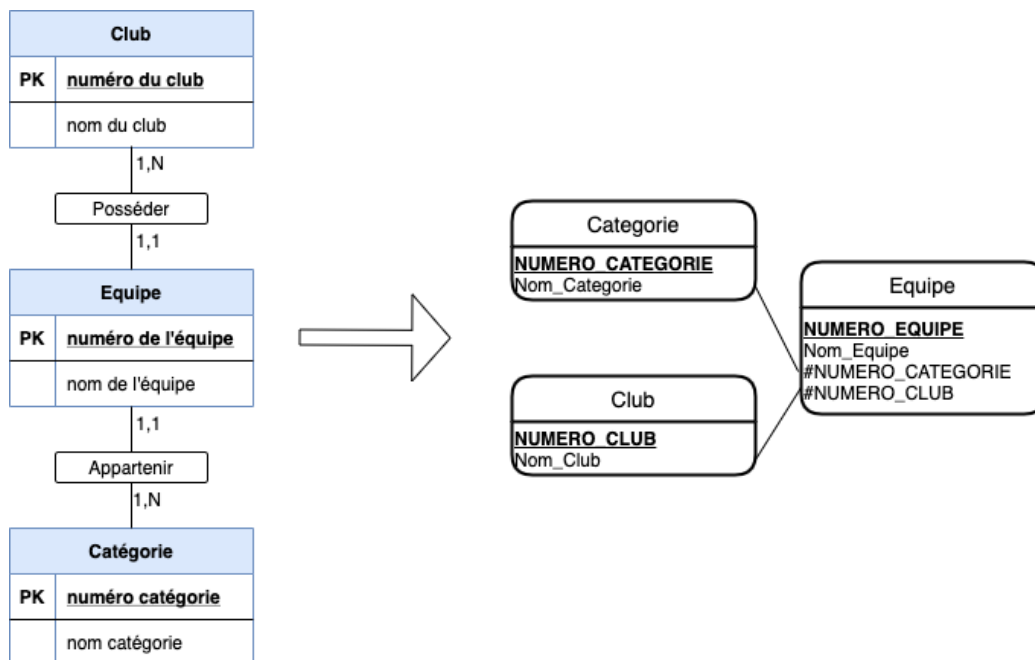


Figure 3 – Association de type (1,N)

Règle Association de type (N,M)

Une association binaire de type (N,M) devient une table. La clé primaire est composée de deux clés étrangères (qui référencent les deux clés primaires des deux tables en association). Les attributs de l'association deviennent des colonnes de cette nouvelle table:

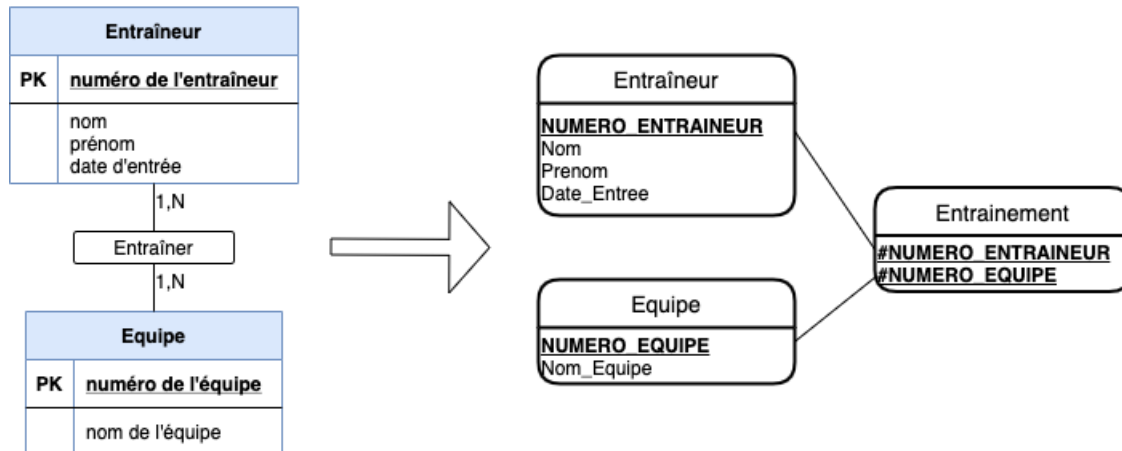


Figure 4 – Association de type (N,M)

Contraintes d'intégrités

Toute les clés primaires des tables possèdent la clause **NOT NULL**. Lors de l'insertion d'une donnée, le système de gestion de base de données refusera donc tout tuple dont le champ d'une clé primaire n'a pas été renseigné . De même que les clés étrangères qui figurent dans la table dont le coté a pour cardinalité (0,1) ou (1,1) comme soulevé dans la règle association de type (1,N).

3.2 Modèle Relationnel final

En appliquant les règles ci-dessus au schéma entité-association nous avons obtenu le modèle relationnel suivant:

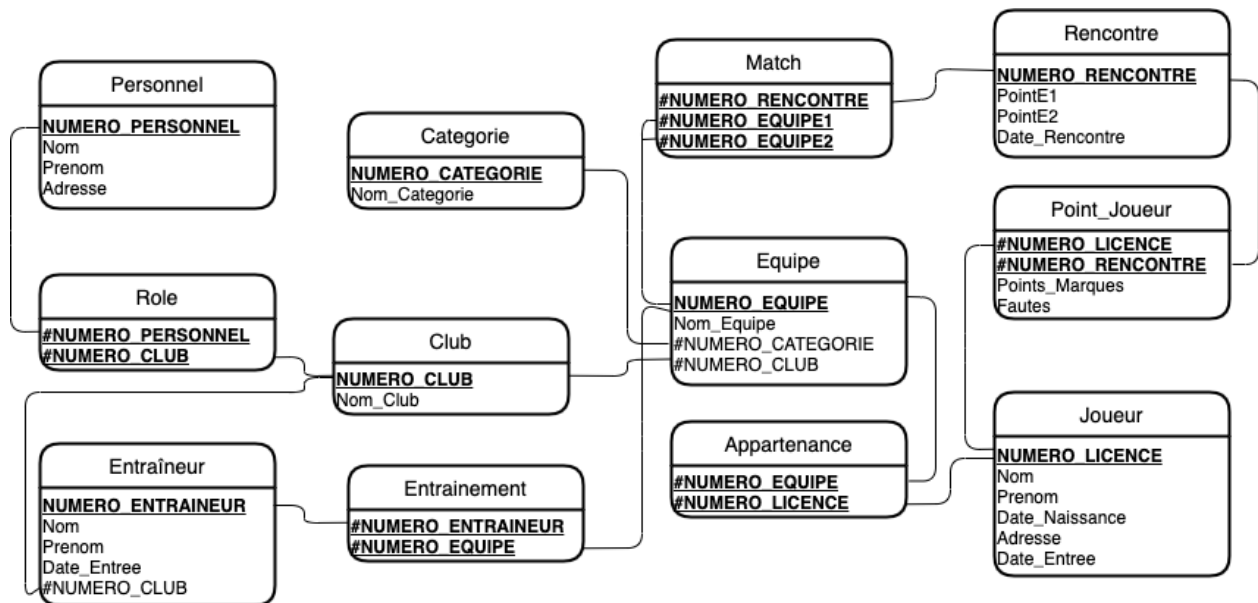


Figure 5 – Modèle relationnel final

Nous pouvons voir à partir de ce modèle que les trois formes normales sont bien respectées:

- Les attributs de toutes les tables ne prennent qu'une seule valeur;
- Tous les attributs non clé d'une table dépendent entièrement de la clé primaire et non d'une partie de la clé.
- Tous les attributs d'une table dépendent uniquement des attributs clé et non des autres attributs de la table.

4 Implémentation de la base de données

4.1 Choix du SGBD

Pour développer ce projet, nous nous sommes intéressés à l'outil **Lamp**, qui comprend un accès à un serveur local, à **MySQL**, ainsi qu'à **PhpMyAdmin**. De ce fait nous avons accès à tous les outils nous permettant de visualiser et implémenter notre base de données, puis d'y ajouter une interface en **PHP** sur notre serveur local. Plusieurs membres du groupe ayant déjà utilisé ces outils par le passé, ce choix était pour nous le plus efficace.

4.2 Requêtes et fonctionnalités

Une requête intéressante à expliquer est celle concernant le classement des équipes. Pour établir ce classement, nous avons considéré que le classement se ferait en fonction du nombre de points marqués par les équipes lors des matchs. Ainsi, la meilleure équipe sera celle qui aura marqué le plus de points. Afin de pouvoir l'analyser, voici la requête en question :

```
SELECT EQUIPE.NOM.EQUIPE, SUM(PE1)
FROM(
SELECT EQUIPE.NOM.EQUIPE AS Nom, SUM(RENCONTRE.POINTE1) AS PE1
FROM (EQUIPE LEFT OUTER JOIN PARTIE
ON EQUIPE.NUMERO.EQUIPE = PARTIE.NUMERO.EQUIPE1)
INNER JOIN RENCONTRE
ON PARTIE.NUMERO.RENCONTRE=RENCONTRE.NUMERO.RENCONTRE
WHERE RENCONTRE.DATE.RENCONTRE <= (SELECT MAX(RENCONTRE.DATE.RENCONTRE)
FROM RENCONTRE)

GROUP BY EQUIPE.NOM.EQUIPE
UNION
SELECT EQUIPE.NOM.EQUIPE, SUM(RENCONTRE.POINTE2)
FROM (EQUIPE LEFT OUTER JOIN PARTIE
ON EQUIPE.NUMERO.EQUIPE = PARTIE.NUMERO.EQUIPE2)
INNER JOIN RENCONTRE
ON PARTIE.NUMERO.RENCONTRE=RENCONTRE.NUMERO.RENCONTRE
WHERE RENCONTRE.DATE.RENCONTRE <= (SELECT MAX(RENCONTRE.DATE.RENCONTRE)
FROM RENCONTRE)

GROUP BY EQUIPE.NOM.EQUIPE) AS TOTAL ,EQUIPE
WHERE EQUIPE.NOM.EQUIPE=Nom
GROUP BY EQUIPE.NOM.EQUIPE
ORDER BY 2 DESC;
```

Cette requête est divisible en 3 étapes, tout d'abord nous avons cherché à récupérer le nombre de points marqué par les équipes lors des rencontres. Il faut savoir que lors d'un match, une équipe peut être enregistré comme **EQUIPE1** ou **EQUIPE2**. Ainsi il va être important de prendre en compte les 2 cas dans le calcul des points. Pour revenir au calcul des points de l'**EQUIPE1**, nous avons fait une somme sur les points marqués par équipe afin d'obtenir en résultat un tableau avec une unique colonne comportant les points de chaque équipe. Afin de s'assurer que même les équipes n'ayant marqué aucun point soient dans le classement nous avons utilisé une jointure externe gauche entre les tables **EQUIPE** et **PARTIE**.

Dans un second temps, nous avons exécuté la même requête que précédemment, mais avec le calcul des points pour l'**EQUIPE2**. Mais pour rassembler le résultat de ces 2 requêtes dans une seule colonne d'un même tableau nous avons fait une **UNION** entre ces 2 requêtes. De ce fait, chaque équipe apparaît 2 fois dans ce tableau de résultat : une fois avec les points marqués en tant qu'**EQUIPE1**, et une autre fois avec les points marqués en tant qu'**EQUIPE2**.

Enfin pour arriver au classement final souhaité, nous avons fait une somme sur le résultat de cette union, afin d'additionner les points marqués par les équipes en tant que 1 et 2, et ainsi obtenir le nombre de point marqué au total. Le premier du classement étant l'équipe ayant marqué le plus de points depuis le début des rencontres.

Du côté des fonctionnalités, nous n'avons pas eu besoin de vue, procédure ou déclencheur. Mais pour assurer la cohérence de notre base de données, lors d'une suppression d'équipe par exemple, nous avons utilisé la fonctionnalité **ON DELETE CASCADE** qui permet d'assurer la suppression de tout ce qui est lié à l'équipe supprimée de la base de données.

5 Interface graphique

Notre interface graphique a été réalisé à l'aide des langages HTML/CSS , PHP, SQL et aussi le framework Bootstrap 4 pour permettre à notre site web d'avoir un design responsive qui s'adapte à tout type d'écran.

La page d'accueil

La page d'accueil contient une brève présentation des différentes opérations pouvons être effectuées sur la base de données. Un menu principal est mis à la disposition de l'utilisateur pour pouvoir manipuler les données de la base comme on peut le voir sur la figure ci dessous.



Figure 6 – Page d'accueil du site web

Le champ consultation

La consultation d'une table ou d'une donnée se fait via le champ 'Consultation' du menu principal. Arrivé sur la page, une liste déroulante de données et de date est proposé à l'utilisateur et ce dernier se voit contraint de choisir au moins une donnée qu'il désire voir.

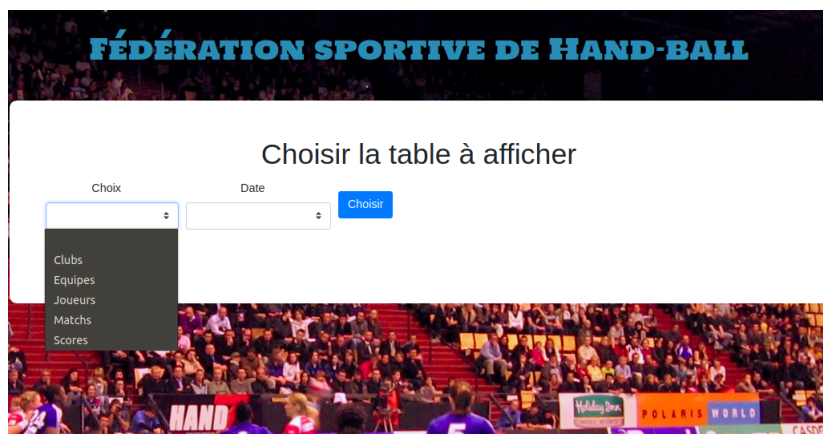


Figure 7 – Page de consultation

Le choix de la date n'est pas obligatoire à moins qu'il veuille consulter la donnée à une date précise. Par exemple, un utilisateur peut choisir de consulter les équipes qui jouent à une date donnée. Si jamais une date n'est pas mentionnée ce sont toutes les équipes qui seront affichés.

Un exemple est illustré sur la figure ci dessous.



Figure 8 – Consultation d'une Equipe

Comme on peut le voir sur la figure il est possible de modifier et de supprimer chaque donnée d'une table.

Le champ insertion

L'interface permet également d'insérer des données dans la base. En cliquant sur le champ, une liste d'éléments est proposée à l'utilisateur comme on peut le voir sur la figure ci dessous.



Figure 9 – Page d’insertion

Après avoir choisit un élément, différents champs avec d’éventuelles listes déroulantes sont proposés à l’utilisateur en fonction de la donnée qu’il veut ajouter. La figure suivante illustre l’ajout d’une nouvelle équipe.

Figure 10 – Ajout d’une nouvelle équipe

Les feuilles de matchs

Pour consulter les différentes feuilles de match, l'utilisateur se verra demander un numéro de rencontre qu'il choisira parmi une liste déroulante de numéros de rencontre disponibles dans la base et qui correspond à la feuille de match qu'il veut consulter.

Sur la feuille de match apparaît le nom des deux équipes qui s'affrontent, les clubs auxquelles elles appartiennent, les joueurs qui ont participé au match ainsi que les points marqués et les fautes de chaque joueur. À la fin, le score du match est affiché.

Les statistiques

L'utilisateur peut également consulter:

- la moyenne des points marqués jusqu'à une date donnée ou du début de la saison jusqu'à la fin;
- les meilleurs joueurs d'une catégorie et à une date donnée qu'il entrera. Si une date n'est pas entrée, c'est le classement des joueurs durant toute la saison qui se fait.
- le classement des équipes.

Le champ résultat

Ce champ permet de voir le nombre de match gagnés, perdus ou nuls par chaque club depuis le début de la saison.



The screenshot shows a web application interface for the 'FÉDÉRATION SPORTIVE DE HAND-BALL'. Below the header, there is a title 'Nombre de Matches gagnés, perdus ou nuls pour chaque Club'. Underneath this title is a table with four columns: 'Nom Club', 'Nombre de Victoire', 'Nombre de défaite', and 'Nombre de Match nul'. The table contains four rows of data for different clubs.

Nom Club	Nombre de Victoire	Nombre de défaite	Nombre de Match nul
EHB EYSINES HANDBALL	3	5	2
GIRONDAIS DE BORDEAUX	3	6	3
MHB MERIGNAC HANDBALL	6	2	4
PHB PESSAC HANDBALL	6	5	1

Figure 11 – Les résultats

6 Conclusion

Ce projet nous a été d'une très grande utilité vis-à-vis de l'application des notions de système de base de données. Nous avons notamment mis en oeuvre une nouvelle fois la conception d'un modèle conceptuel de données ainsi que l'application des règles de traduction de celui-ci en modèle relationnel de données. La réalisation d'une interface graphique a permis une manipulation plus simple et directe des données de la base.

Ce projet a aussi été l'occasion d'affiner notre habilité à travailler en équipe, dans des conditions inhabituelles puisqu'il a été programmé durant le confinement à cause de la pandémie du Covid-19.