



Filière informatique
ENSEIRB-MATMECA

— PROJET SGBD —

Fédération sportive de Hand-ball

Asmae ZIANI Zaid ZERRAD Saad ZOUBAIRI Hicham
ZGHARI

Encadré par Sylvain LOMBARDY

Table des matières

1	Introduction	3
2	Modèle conceptuel	3
3	Schéma relationnel	3
4	Implémentation	5
4.1	Choix du SGBD	5
4.2	Requêtes remarquables	5
4.2.1	Classement des équipes	5
4.3	Déclencheurs de vérification	6
5	Interface	6
5.1	Choix du type d'interface	6
5.2	Utilisation	6
5.3	Fonctionnalités	7
5.4	Consultation	7
5.4.1	Insertion	8
5.4.2	Classement	10
5.4.3	Faire votre test	11
6	Conclusion	12
7	Annexes	12

1 Introduction

Ce projet consiste à mettre en oeuvre les apprentissages et les notions vues en module de SGBD. Les étapes du projet consiste à passer par toutes les étapes de la réalisation d'une base de données, commençant par les modélisation de données et la création d'une base de données relationnelle et finissant par l'implémentaion de quelques opérations pour consulter ou modifier les données de la base.

Le sujet proposé pour ce projet, est la réalisation d'une base de données permettant de gérer les rencontres de Hand _ ball entre les clubs d'une fédération. Plusieurs informations sont fournis pour la réalisation de la base à savoir : l'administration au sein d'un club, la catégorisation des équipe d'un club, la gestion des joueurs et des entraîneurs, et l'enregistrement des rencontres. Mais l'étude de ces informations nous a poussé à apporter quelques ajouts à la version initiale, ces ajouts seront justifiées dans la section suivante et illustrés par le modèle conceptuel associé.

2 Modèle conceptuel

Plusieurs suppléments ont été ajouté au sujet proposé. A titre d'exemple, l'information qui situe la rencontre dans un cadre temporel à savoir la saison a été ajouté comme entité dans le modèle conceptuel proposé, et qui est en association avec le joueur. De plus, la gestion des stades possédés par les clubs et qui accueillent les rencontres est représentée dans notre modèle par l'entité stade qui est de ce fait en association avec les entités club et rencontres. Le modèle qui résume la totalité de l'étape de conception de données envisagée est illustré par la figure de l'annexe.

Cette modélisation des données représente une évolution par rapport au modèle proposé au premier rendu du fait que la catégorisation des équipes est maintenant prise en compte grâce aux deux champs de l'entité **Equipe** (**Niveau** et **Catégorie**). De plus grâce à l'association binaire de cardinalités maximale **n:n** entre **JOUEUR** et **EQUIPE** nous pouvons gérer l'historique de l'appartenance d'un joueur à une date donnée à une équipe donnée, ceci se fait notamment grâce aux attributs de l'association en question **JOUER** (**date_entree** et **date_sortie**). De plus, un autre problème qui pourrait se poser est qu'un certain joueur puisse quitter et rejoindre la même équipe, ce problème est résolu grâce à une convention que nous avons fixée sur la date de sortie (prise égale à 9999) si le joueur est toujours dans l'équipe considérée.

3 Schéma relationnel

Après l'étape de conception de données, nous avons traduit notre modèle conceptuel en schéma relationnel qui est le suivant :

- CLUB(ID_CLUB, NOM_CLUB, NUMERO_TELEPHONE, ADRESSE_CLUB, DATE_CREATION)
- RESPONSABLE(ID_RESPONSABLE, #ID_CLUB, NOM_RESPONSABLE, PRENOM_RESPONSABLE, DATE_NAISSANCE, ADRESSE_RESPONSABLE, NATIONALITE, FONCTION)

- EQUIPE(ID_EQUIPE, #ID_CLUB, NOM_EQUIPE, CATEGORIE, NIVEAU, DATE_CREATION_EQUIPE)
- ENTRAINEUR(NUMERO_LICENCE_ENTRAINEUR, NOM_ENTRAINEUR, PRENOM_ENTRAINEUR, DATE_NAISSANCE_ENTRAINEUR, NAIONALITE_ENTRAINEUR, ADRESSE_ENTRAINEUR)
- ENTRAINER(#ID_EQUIPE, #NUMERO_LICENCE_ENTRAINEUR, DEBUT_CONTRAT, FIN_CONTRAT)
- JOUEUR(NUMERO_LICENCE_JOUEUR, NOM_JOUEUR, PRENOM_JOUEUR, DATE_NAISSANCE_JOUEUR, NATIONALITE_JOUEUR, ADRESSE_JOUEUR)
- JOUER(#ID_EQUIPE, #NUMERO_LICENCE_JOUEUR, DATE_ENTREE, DATE_SORTIE)
- RENCONTRE(ID_RENCONTRE, #NOM_STADE, DATE_RENCONTRE, HEURE_RENCONTRE)
- RESULTATS_RENCONTRES(#ID_EQUIPE, #ID_RENCONTRE, SCORE_EQUIPE)
- PRESTATION_JOUEUR(#ID_RENCONTRE, #NUMERO_LICENCE_JOUEUR, POINTS_MARQUES, FAUTES_COMMISES)
- STADE(NOM_STADE, #ID_CLUB, VILLE_STADE, CAPACITE, SURFACE)
- SAISON(ID_SAISON, DATE_DEBUT_SAISON, DATE_FIN_SAISON)
- PARTICIPATION(#ID_SAISON, #NUMERO_LICENCE_JOUEUR)

On peut illustrer ces relations de manière plus visuelle comme dans la figure suivante :

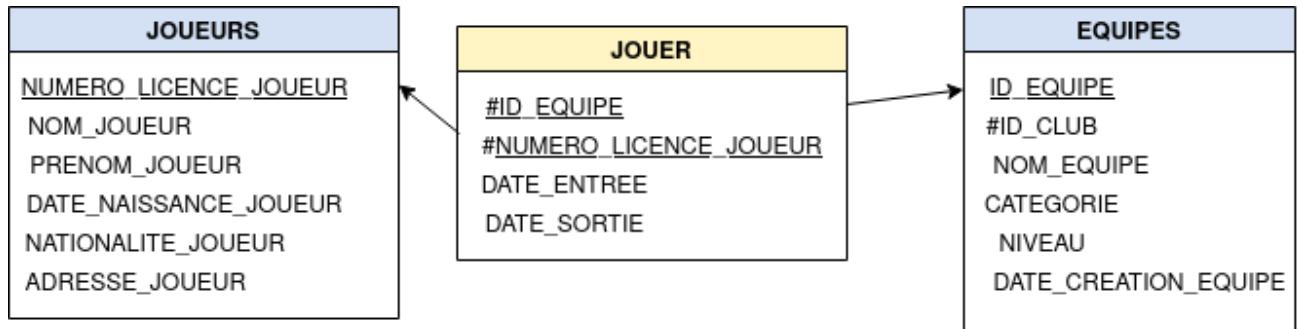


FIGURE 1 – Illustration des relations JOUEURS, JOUER et EQUIPES

Ce modèle proposé vérifie les trois formes normales :

1. Aucun attributs ne contient une liste de valeurs
2. Tous les champs de nos entité dépendent de la clé primaire en entier
3. Tous les champs des entités du schéma dépendent uniquement et directement des identifiants des tables.

4 Implémentation

4.1 Choix du SGBD

Étant familiarisés avec **Oracle** durant les TPs, nous l'avons choisi pour l'implémentation de notre base de données, et nos requêtes. Nous avons tout de même trouvé un problème lors de l'implémentation de l'interface en **php**, nous avons décidé alors de passer à **Mysql**. Nous avons fini donc par garder les deux implémentations dans deux réertoires séparés que vous trouverez dans notre rendu (la première avec **Oracle** la seconde avec **Mysql** et qui contient les fichiers de l'interface utilisée).

4.2 Requêtes remarquables

Le fichier **Oracle/requettes.sql** contient l'implémentation de plusieurs requêtes qui permettent de chercher, mettre à jour, comparer, ou insérer une information dans la base de données. Dans ce qui suit, on présentera les requêtes implémentées les plus importantes.

4.2.1 Classement des équipes

Dans le but d'implémenter une requête qui affiche le classement des équipes lors d'une saison donnée, nous avons tâché à définir deux vues intermédiaires, **SS** et **CLASSEMENT**. En effet, la première vue, **SS**, consiste à créer une table qui contient l'identifiant de chaque rencontre jouée lors de la saison (on élimine les rencontres qui ne sont pas encore jouées si l'identifiant de la saison coïncide avec la saison liée à l'année actuelle.), les identifiants des deux équipes confrontées et le nombre de buts marqués par chaque équipe. La deuxième vue ,**CLASSEMENT**, contient 3 champs qui permettent le regroupement d'informations, ainsi que 3 champs (**nb_matchs_gagnes**, **nb_matchs_nuls** et **nb_matchs_perdus**) qui permettent le calcul du résultat de la rencontre. En effet, pour une ligne donnée, nous avons 3 cas à étudier :

- Si le nombre de buts marqués par la première équipe est supérieur à celui de la deuxième équipe, la colonne **nb_matchs_gagnes** prend 1 sinon **null** (se représente par un 0 dans l'affichage).
- Si le nombre de buts marqués par les deux équipes sont égaux alors la colonne **nb_matchs_nuls** prend 1, sinon **null**.
- Si le nombre de buts marqués par la première équipe est inférieur à celui de la deuxième équipe, la colonne **nb_matchs_perdus** prend 1 sinon **null** (se représente par un 0 dans l'affichage).

Ainsi pour chaque ligne, nous aurons l'information sur le résultat du match (gagné, nul ou perdu) ainsi que les buts marqués et concédés par l'équipe (utilisés plus tard pour le calcul de la différence des buts).

Par suite, notre requête principale qui affiche le classement se base sur le résultat de **CLASSEMENT**. En effet, nous réalisons un groupement par identifiant d'équipe et nous

calculons le nombre de matchs gagnés (resp nuls, perdus) en faisant une somme sur la colonne `nb_matchs_gagnés` (resp `nb_matchs_nuls`, `nb_matchs_perdus`). Ensuite, nous calculons la différence de buts en faisant la somme sur la différence entre le nombre de buts marqués et concédés de toutes les lignes du groupement ($\sum(score1 - score2)$). L'information principale à connaître est le nombre de points obtenus, ainsi, nous le calculons à travers la relation $2 * match_gagnes + 1 * matchs_nuls$. Le critère principal de classement des équipes est le nombre de points suivi de la différence de buts. Voici un exemple de l'affichage résultant :

ID_EQUIPE	MATCHS_GAGNES	MATCHS_NULS	MATCHS_PERDUS	DIFF	PTS
1	2	0	0	17	4
3	1	0	1	-5	2
2	0	0	2	-12	0

FIGURE 2 – Affichage de la requête de classement des équipes

4.3 Déclencheurs de vérification

Lors de l'insertion d'une équipe, ou d'un joueur à une équipe (la table `jouer`), on lance un déclencheur qui vérifie l'intégrité des données entrées.

Pour une nouvelle **équipe**, sa date de création doit être supérieure ou égale à la date du **club** auquel elle appartient.

De même, pour un joueur qui commence à jouer dans une équipe (ajout à la table **jouer**), sa date de sortie doit être supérieure strictement à sa date d'entrée. Il doit aussi ne pas jouer pour deux équipes en même temps, il faut donc vérifier que s'il a déjà jouer pour d'autres équipes que ses dates de sortie de ces équipes soient toutes inférieures à sa date d'entrée dans la nouvelle équipe.

5 Interface

5.1 Choix du type d'interface

Notre interface est une interface Web écrite principalement en PHP/HTML/CSS avec l'utilisation de MySQL et phpMyAdmin afin de générer la base de données.

5.2 Utilisation

Dans cette partie, nous allons décrire les étapes à suivre afin d'accéder à notre interface sur votre machine Linux.

1. Installer le paquetage XWAMP (la combinaison Apache + PHP + MySQL) .

2. lancez les commandes suivantes :
- ```
sudo su
/opt/lampp/lampp start
```
3. Placer le répertoire PHP dans le dossier /vv/www/html/
  4. Exécuter le fichier index.html.
  5. Bienvenue à notre interface.

### 5.3 Fonctionnalités

Notre interface Web contient les fonctionnalités nécessaires pour accéder à notre base de données, ainsi que d'exécuter certaines requêtes qui permettent d'afficher des statistiques liées à nos tables (Club - Equipe - Joueur ...). La page d'accueil de notre interface Web contient les fonctionnalités illustrées par la figure suivante :



FIGURE 3 – Page d'accueil

### 5.4 Consultation

Cette fenêtre permet d'accéder aux informations stockées dans notre base de données. Cet accès se fait par un choix de table comme suit :

The figure consists of two screenshots of a web application. The top screenshot shows a consultation form with a green 'Submit' button. The bottom screenshot shows a table of club data with 12 rows, each containing information such as ID, club name, phone number, address, and creation date.

| ID CLUB | NOM CLUB     | NUMERO TELEPHONE | ADRESSE CLUB                     | DATE CREATION |
|---------|--------------|------------------|----------------------------------|---------------|
| 1       | WYDAD AC     | 600000000        | Avenue des Sports Casablanca     | 1937-05-08    |
| 2       | RAJA AC      | 600000001        | Avenue des Sports Casablanca     | 1949-01-01    |
| 3       | MAS          | 600006811        | Fes                              |               |
| 4       | FC BARCELONE | 912345678        | Barcelone                        | 1942-04-08    |
| 5       | REAL MADRID  |                  | Madrid                           | 1902-03-06    |
| 6       | LIVERPOOL HC | 656561212        | Liverpool                        | 1968-11-28    |
| 7       | PSG HC       | 601010000        | 10 Rue les Tulipes Paris         | 1995-02-24    |
| 8       | Montpellier  | 614725836        | Avenue des Cheverues Montpellier |               |
| 9       | Fuchse HC    | 811111111        |                                  | 1998-06-18    |
| 10      | ESS HC       | 600010002        | Sousse                           | 1965-05-25    |
| 11      | AL AHLY SC   | 612378945        | Cairo                            | 1959-12-12    |
| 99      | Test         | 558596320        | local adress                     | 1999-12-31    |

FIGURE 4 – Consultation

#### 5.4.1 Insertion

Nous avons ajouté également une fenêtre d'insertion afin de permettre aux utilisateurs d'ajouter des informations dans la base de données. La figure suivante explique

la méthode suivie :

The figure consists of two screenshots of a web application. The top screenshot shows a main insertion page with three nested forms. The innermost form is titled 'CLUB' and contains fields for 'EQUIPE' and 'JOUEUR'. The middle form is titled 'EQUIPE'. The outermost form is titled 'JOUEUR'. The bottom screenshot shows a detailed 'Club' insertion form with fields for ID\_CLUB, NOM\_CLUB, NUMERO\_TELEPHONE, ADRESSE\_CLUB, and DATE\_CREATION, each with its own input field. A green 'Submit' button is at the bottom.

Federation Sportive/Insertion

localhost/mymysql/web/insertion.php?

# Insertion

CLUB

EQUIPE

JOUEUR

Federation Sportive/Insertion

localhost/mymysql/web/insert\_club.php?

# Insertion

ID\_CLUB:

NOM\_CLUB:

NUMERO\_TELEPHONE:

ADRESSE\_CLUB:

DATE\_CREATION:

Submit

FIGURE 5 – Insertion

#### 5.4.2 Classement

Cette fenêtre est utile pour afficher le classement des équipes pendant une saison données, ainsi que le classement des joueurs par rapport à une catégorie choisie et une date précise :

The figure consists of two screenshots of a web application interface. Both screenshots have a header bar with a back/forward button, a search bar containing 'localhost/mymysql/web/classement.php?', and a green 'Submit' button.

**Top Screenshot (General Ranking):**

- Table:** A dropdown menu.
- CATEGORIE:** A dropdown menu.
- SAISON:** A dropdown menu.
- DATE (par ex: 1999-12-31):** An input field.
- Submit:** A green button.

**Bottom Screenshot (Specific Ranking):**

- Table:** A dropdown menu set to 'JOUEUR'.
- CATEGORIE:** A dropdown menu set to 'JUNIOR'.
- SAISON:** A dropdown menu set to '2019-2020'.
- DATE (par ex: 1999-12-31):** An input field set to '2019-04-01'.
- Submit:** A green button.

**Results Table:**

| Classement | Id_Joueur | Nom_Joueur | Points | Fautes |
|------------|-----------|------------|--------|--------|
| 1          | 1100      | HAKIMI     | 5      | 1      |
| 2          | 100000    | BASIK      | 4      | 1      |
| 3          | 100001    | ZAKI       | 4      | 2      |

FIGURE 6 – Classement

### 5.4.3 Faire votre test

Finalement, si nous avons oublié une fonctionnalité à afficher, nous invitons les utilisateurs à exécuter leurs commandes à travers la fenêtre suivante :

The image consists of two vertically stacked screenshots of a web browser window titled "Federation Sportive/Faire".

**Screenshot 1 (Top):** A query input form. The URL in the address bar is "localhost/mymysql/web/tester.php?". The "Requete:" field contains the SQL query: "select \* from CLUB where ID\_CLUB=2;". Below the input field is a green button labeled "Submit Query".

**Screenshot 2 (Bottom):** The results of the query. The URL in the address bar is "localhost/mymysql/web/test.php". The results are displayed in a table:

| > > > > > |         |           |                                 |            |
|-----------|---------|-----------|---------------------------------|------------|
| 2         | RAJA AC | 600000001 | Avenue des Sports<br>Casablanca | 1949-01-01 |

FIGURE 7 – Tester

## **6 Conclusion**

Ce projet de création et d'exploitation d'une base de données a été une très bonne opportunité pour mettre en pratique tous les concepts vus en cours de SGBD et de passer par toutes les étapes de réalisation d'une base de données allant de la conception jusqu'à la réalisation. Ceci s'est concrétisé à travers un modèle intéressant qui nous a confronté à plusieurs problématiques à savoir la gestion de l'historique des joueurs dans les équipes. De plus, nous avons constaté que pour des moyens d'optimisation (pour éviter de calculer à chaque fois un attribut qu'on consulte souvent) nous pourrions tolérer des redondances au niveau des attributs (attributs calculables à partir d'autres attributs).

## **7 Annexes**

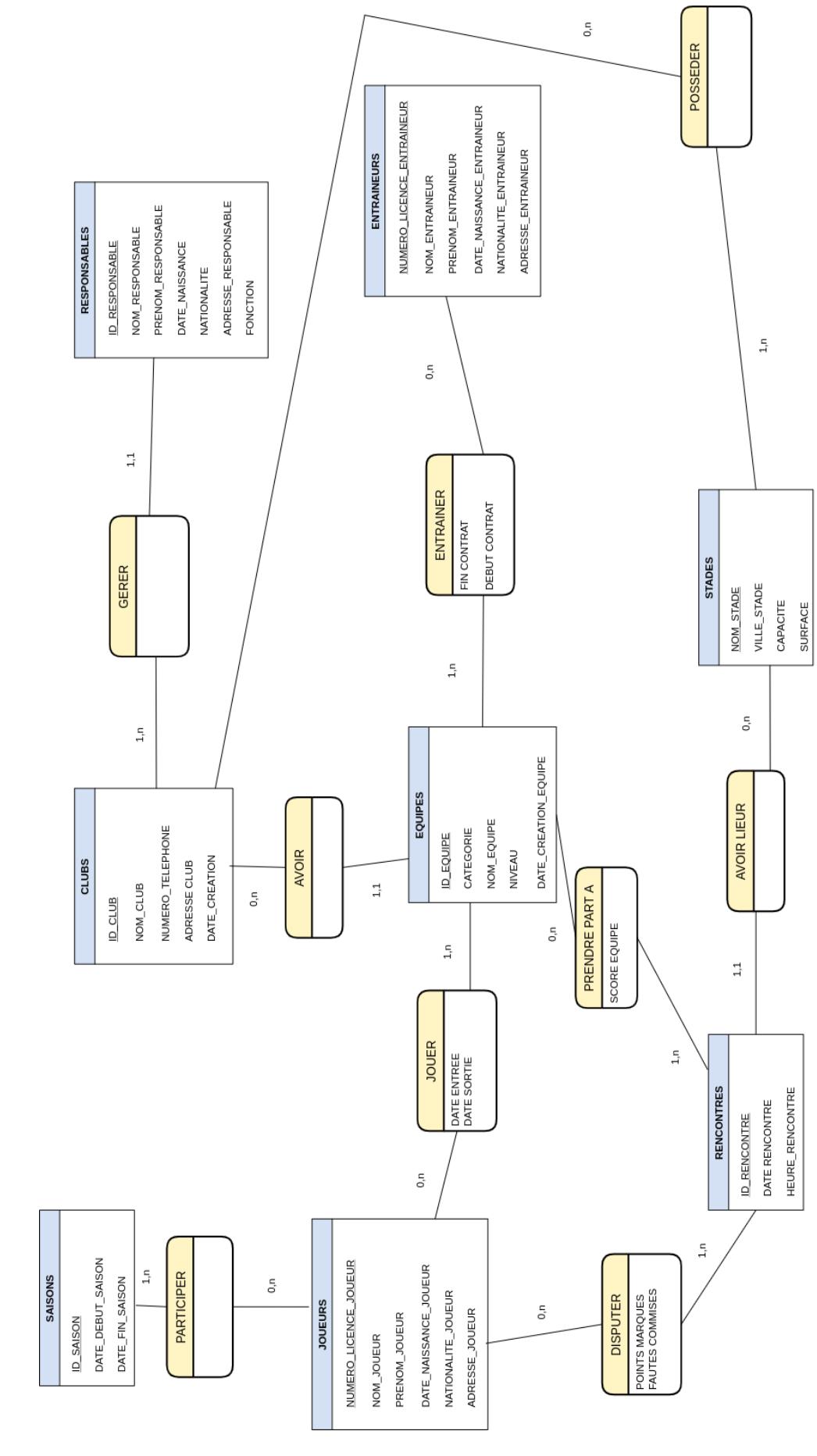


FIGURE 8 – Modèle conceptuel des données