

UNIVERSITÉ DE MONTPELLIER
L3 INFORMATIQUE

Base de données Tournoi Sportif

RAPPORT DE PROJET
SYSTÈMES D'INFORMATION ET BASES DE DONNÉES 2 - HLIN511
ANNÉE 2020-2021
GROUPE A3

Étudiants

Merwane RAKKAOUI (n°21914058)

Ali DABACHIL (n°21719405)



Table des matières

1	Introduction	3
2	Organisation du projet	4
2.1	Outils utilisés	4
3	Élaboration du Modèle Entité-Association	5
3.1	Aperçu du modèle Entité-Association	5
3.2	Explications du Modèle E/A proposé	6
3.3	Passage au modèle relationnel	7
3.4	Mise en place des procédures, fonctions et triggers associés . . .	8

Chapitre 1

Introduction

Ce projet s'inscrit dans le cadre du développement d'une application web permettant d'assurer la gestion de tournois sportifs en ligne. Cette application web concerne une partie de ce projet, partie n'ayant pas été traitée.

Nous nous intéresserons principalement ici à la partie Base de données sur la gestion de tournois dans le cadre de ce projet, qui s'articulera autour de plusieurs axes.

Il sera important d'aborder une étude conceptuelle pour ce projet à travers l'élaboration du modèle Entité-Association (E/A) propre au problème étudié. Une telle élaboration nous permettra par la suite d'obtenir un modèle relationnel.

Une partie programmation viendra compléter l'étude conceptuelle, elle traitera de l'explication des procédures, fonctions et triggers associés, accompagnées de tests associés.

Chapitre 2

Organisation du projet

2.1 Outils utilisés

Pour faire ce projet nous avons utilisé :

phpMyAdmin, une application web développée en langage PHP ayant pour objectif de s'occuper de l'administration d'un serveur de base de données MySQL ou MariaDB. Son environnement est multi-plateforme. phpMyAdmin propose une multitude de fonctionnalités telles que la création de tables, vues, index ou encore l'exécution de requêtes SQL et bien d'autres encore. Cet outil a été très important notamment pour la création de la base de données tournoiSportif.

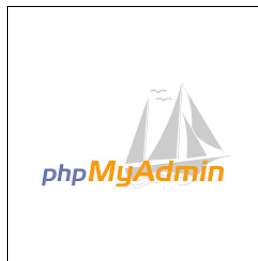


FIGURE 2.1 – logo phpMyAdmin

Looping, un logiciel de modélisation conceptuelle de données développé par l'Université Toulouse III. Celui-ci permet d'organiser et de structurer des données en diagrammes (E/A, UML et MLD). Ce logiciel offre la possibilité d'exécuter des requêtes SQL ou encore la création de tables à partir d'un script SQL. Une élaboration du modèle Entité-Association a été permise par l'intermédiaire de ce logiciel.

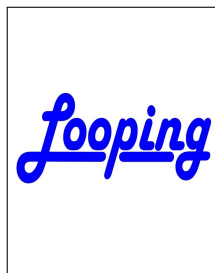


FIGURE 2.2 – logo Looping

Chapitre 3

Élaboration du Modèle Entité-Association

3.1 Aperçu du modèle Entité-Association

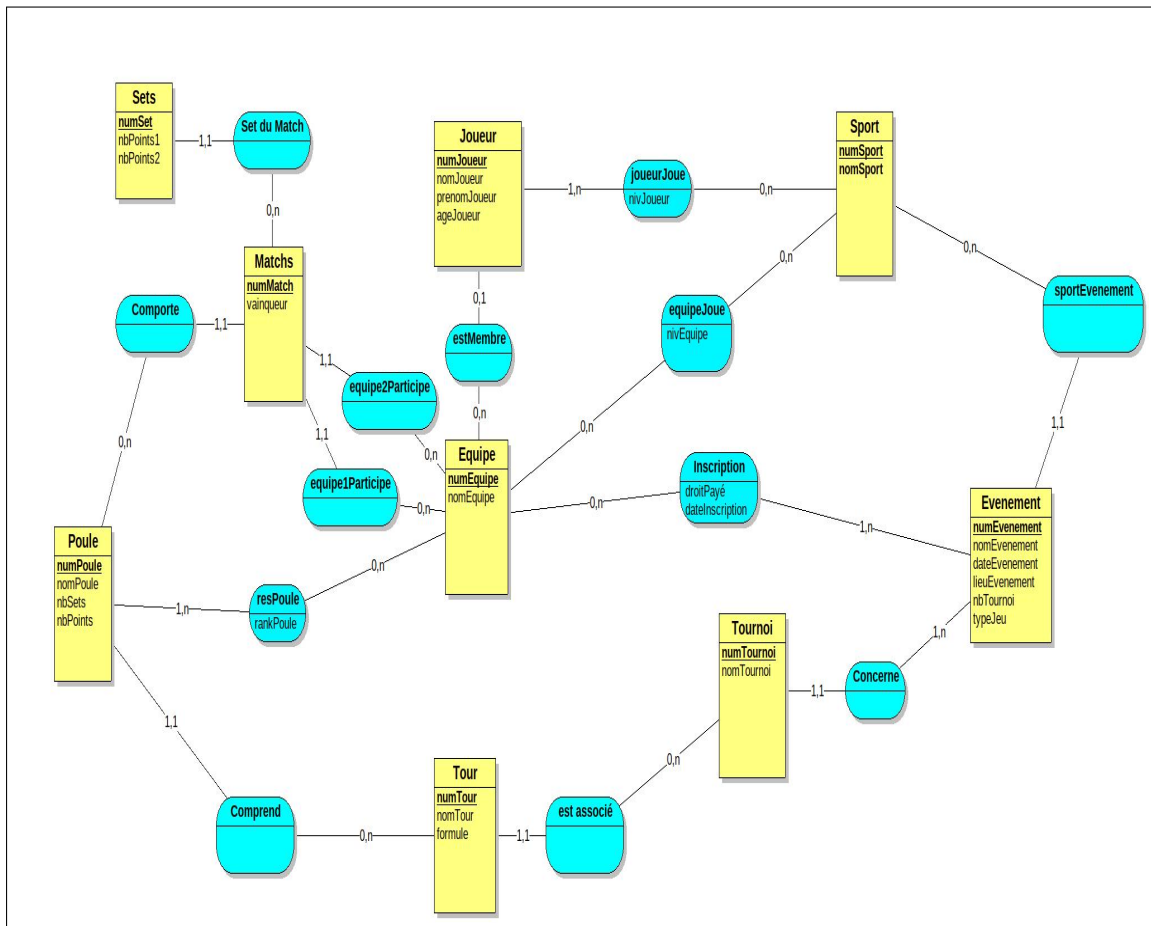


FIGURE 3.1 – Tournoi Sportif, modèle Entité-Association

3.2 Explications du Modèle E/A proposé

Tout d'abord, un événement identifié par sa clé primaire **numEvenement** concerne moins un tournoi sportif. Il spécifie un lieu, une date, un type de jeu(2 joueurs, 3 joueurs, 4 joueurs), ainsi que le nombre et le noms de ses tournois initiaux(féminin, loisir, jeune, moins de 17 ans, ...). Un tournoi sportif se caractérise par un matricule **numTournoi** et un nom. Dans un même tournoi, il peut y avoir plusieurs tournois (tournoi principale, tournoi consolante).

Un joueur caractérisé par son numéro de joueur **numJoueur**, son nom, son prénom ainsi que son âge. Il est membre d'au plus une équipe. Il a la possibilité de pouvoir pratiquer plusieurs sports dans notre conception avec n'importe quel niveau, son niveau est propriété de l'association "joueurJoue". Le niveau présenté figure parmi la liste suivante :

- Loisir
- Départemental
- Régional
- N3
- N2
- Elite
- Pro

Une équipe caractérisée par un numéro d'équipe **numEquipe** et un nom, compte un certain nombre de joueurs. Pour qu'une équipe puisse participer à un événement, il est nécessaire que celle-ci s'acquitte des droits d'inscription. Les droits d'inscriptions ajoutés à la date de l'inscription constituent la propriété de l'association "Inscription" entre les entités Equipe et Evenement. Ceci justifie l'intérêt d'une telle association. Bien sûr, il faudra veiller à ce que la date d'inscription ait lieu avant la date de l'événement.

Une équipe peut jouer à plusieurs sports, le niveau de l'équipe étant propriété de l'association "equipeJoue" sera calculé à partir d'une fonction qui va être abordée plus tard dans ce rapport.

On est parti du principe que le sport pratiqué durant un événement était unique. Étant donné qu'un événement concerne un unique tournoi sportif, un tournoi est propre à un sport parmi la liste suivante [Pétanque, Volley, Tennis, Football]. Un sport est identifié par son numéro **numSport** et son nom.

Lors de la fin d'un tournoi sportif, un résultat de type entier "rankTournoi" sera établi afin de déterminer le placement définitif d'une équipe dans un tournoi. Ce résultat est propriété de l'association liant une équipe à un tournoi.

Dans un même tournoi, il peut y avoir un certain nombre de tours. Un tour est identifié grâce à un numéro **numTour**, un nom de tour (exemple : Tour1, Tour2) et une formule définissant le nombre de poules ainsi que le nombre d'équipes respectives de chaque poule pour un tour donné.

Une poule est un mini-championnat dans laquelle toutes les équipes s'affrontent. Elle peut comporter une série de matchs et comprend un seul et unique tour. Chaque poule est caractérisée par un identifiant **numPoule**, un nom de poule (ex : Poule1, Poule2), un nombre de sets et un nombre de points. Le classement des équipes affectées à une poule donnée peut être déterminé à travers l'association "resPoule" ayant pour propriété "rankPoule".

Deux équipes (équipe1 et équipe2) s'affrontent dans un match, l'entité Matches caractérisée par un numéro **numMatch**, peut déterminer de manière unique le vainqueur de la rencontre.

Lors d'une rencontre, il y a un nombre de sets et de points définis. Le nombre de points

obtenus par chaque s'opposant dans un match peut être obtenu à partir de l'entité Sets. Cette entité est identifiée grâce à un numéro **numSet**, le nombre de points obtenus par l'équipe1 'nbPoints1' ainsi que le nombre de points obtenus par l'équipe2 'nbPoints2'.

3.3 Passage au modèle relationnel

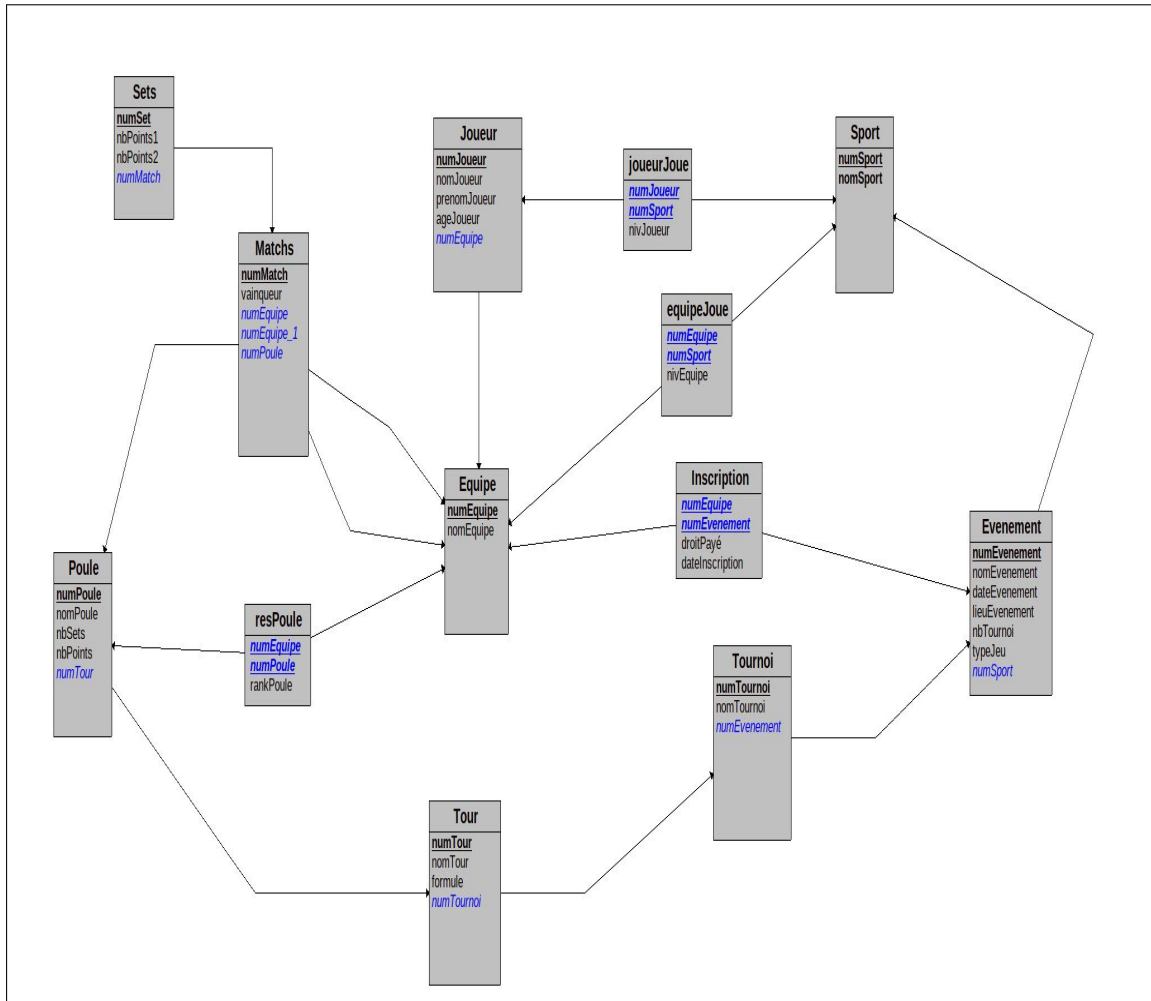


FIGURE 3.2 – Tournoi Sportif, modèle logique de données (relationnel)

3.4 Mise en place des procédures, fonctions et triggers associés

On va commencer à expliquer dans un premier temps les fonctions implémentées qui ont été utiles pour notre base de données. La fonction **getnivE** prend comme paramètres un niveau d'équipe ainsi qu'un numéro de sport. Une équipe comprend une liste de joueurs ayant un niveau parmi la liste suivante : [loisir, départemental, régional, N3, N2, Elite, Pro] On déclare 7 variables (nb1 à nb7) désignant le nombre de joueurs ayant un niveau correspondant à l'un des niveaux présent dans la liste ci-dessus.

Exemple : nb1 -> loisir, nb2 -> départemental, ..., nb7 -> Pro

On compte ensuite le nombre de joueurs ayant un niveau parmi la liste proposée ci-dessus.

De manière linéaire, on parvient à retourner le niveau général d'une équipe en connaissant le niveau de chacun de ses joueurs.

La fonction **getWr** prenant comme paramètre un nom d'équipe 'nomE'. Dans le corps de cette fonction, on détermine le nombre de victoires d'une équipe1 et celui d'une équipe2 s'opposant dans un match. On retourne à la fin le ratio de victoires pour une équipe donnée.

Il est maintenant question d'expliquer la mise en place de nos triggers. Dans le premier trigger **setNivJoueur**, on initialise le niveau d'un joueur à loisir à chaque fois qu'on l'insère un nouveau joueur dans la table joueur.

Le second trigger **updateINSERTnivEquipe** s'exécute après l'insertion d'un nouvel enregistrement dans la table joueurJoue. Le niveau de l'équipe dans la table joueurJoue est mis à jour grâce à la fonction **getnivE** définie plus haut.

Exemple de requête qui affiche la liste des équipes qui ont gagné une grande finale :

```
SELECT DISTINCT CASE WHEN vainqueur = 1 THEN numEquipe1 ELSE numEquipe2 END as Gagnant FROM poule,respoule,matches WHERE poule.numPoule = respoule.numPoule AND matches.numPoule = poule.numPoule AND nomPoule LIKE 'AND nomPoule LIKE '
```