



Projet HLIN511 Groupe I

Base de donnée sur la gestion de tournoi

Rapport rédigé par :
ABRADI Chaima N°etudiant : 21700074
MESSAOUDI Soukaina N°etudiant : 21603069

Table des matières

1	Schéma entité association et relationnel	3
1.1	Schéma entité association	3
1.2	Schéma relationnel	3
2	Langage de programmation procédurale	4
2.1	Première fonction	4
2.2	Seconde fonction	4
2.3	Triggers	4
3	Tests	5
3.1	Tests fonction	5
3.2	Tests Triggers	5

Introduction

Un tournoi sportif est une compétition sportive qui implique l'élimination successive des compétiteurs afin d'obtenir le gagnant. Ce championnat complexe exige une organisation et une gestion minutieuse nécessaire au bon déroulement du tournoi. La gestion d'un tournoi sportif implique la prise en compte de nombreux paramètres. Il advient au début de recueillir les informations principales sur l'événement, les participants et les lieux.

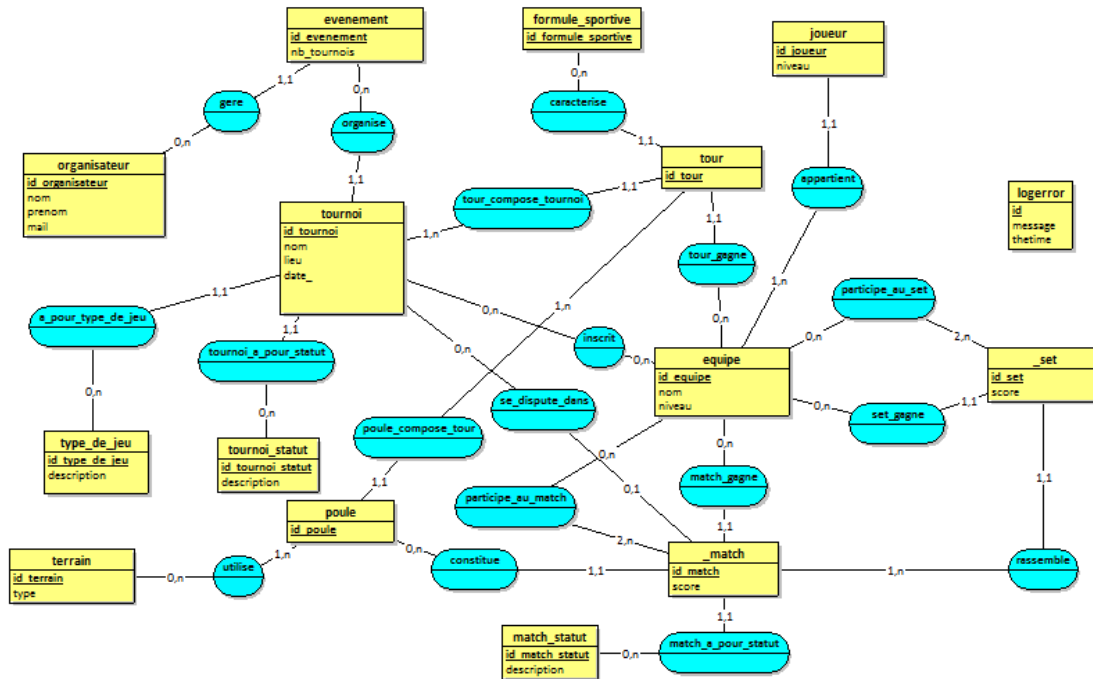
Dans le cadre du module HLIN511 nous avons été en charge de proposer et d'analyser une base de données de gestion de tournoi.

C'est pourquoi nous montrerons dans une première partie le modèle entité-association que nous avons choisi de mettre en place ainsi que son explication et le modèle relationnel associé. Dans une seconde partie nous utiliserons le langage de programmation procédurale à travers des fonctionnalités tels que les triggers et fonctions. Enfin nous expliquerons les tests que nous avons décidé d'effectuer sur ce projet.

1 Schéma entité association et relationnel

1.1 Schéma entité association

Le modèle que nous avons choisis de mettre en place afin de modéliser la gestion du tournoi est le suivant :



Lors de la gestion d'un événement un organisateur gère au maximum n événement, un événement est géré par un et un seul organisateur. Un événement organise n tournoi. Un tournoi organise au maximum un seul événement et possède un seul type de jeu.

Une équipe est composée de plusieurs joueurs, ses joueurs sont uniques ils appartiennent qu'à une seule équipe. Une équipe s'inscrit à n tournois donc un tournoi contient n équipes inscrite d'où la cardinalité (0,n). Après inscription, en fonction du tournoi choisi il est composé de 1 ou plusieurs tour tandis qu'un tour a un et seul tournoi, chaque tour est caractérisé par une seule formule sportive. Un tour est composé de plusieurs poules, alors qu'une poule est composée au maximum d'un seul tour.

Une poule est utilisée par plusieurs terrains de même pour un terrain il est utilisé par plusieurs poules d'où l'apparition de l'association 'utilise' qui va pouvoir par exemple nous affecter pour une poule plusieurs terrains et un terrain plusieurs poules ce qui n'est pas possible de le faire sans une association (d'où la cardinalité 0,n et 1,n).

Une équipe gagne n tours, alors qu'un tour est gagné par une seule équipe, elle gagne aussi n sets.

Une équipe participe à plusieurs sets et un set est participé par plusieurs équipes, d'où l'association 'participe au set'.

Dans un match il y a au moins 2 équipes qui participent. Une équipe participe à n matchs, donc ici nous avons introduit une association 'participe au match' qui pourra affecter à plusieurs équipes un match et plusieurs matchs à une équipe.

Chaque match possède un seul statut tandis qu'un statut est affecté par plusieurs matchs.

Un match rassemble par plusieurs ou un seul set, cependant un set est rassemblé par un seul match.

1.2 Schéma relationnel

Le schéma relationnel correspondant au modèle entité association est le suivant, notons que les clés primaires sont soulignées et que les clés étrangères sont en italiques.

```

organisateur (id_organisateur , nom, prenom , mail );
formule_sportive = (id_formule_sportive );
terrain(id_terrain , type);
equipe = (id_equipe , nom, niveau);
joueur = (id_joueur , niveau , id_equipe);
tournoi_statut = (id_tournoi_statut , description );
match_statut = (id_match_statut , description);
type_de_jeu = (id_type_de_jeu , description );
logerror = (id , message , thetime );
evenement = (id_evenement , nb_tournois , id_organisateur);
tournoi = (id_tournoi, nom, lieu , date_ , id_type_de_jeu, id_tournoi_statut, id_evenement);
tour = (id_tour, id_equipe, id_formule_sportive, id_tournoi);
poule = (id_poule , id_tour);
_match = (id_match , score , id_tournoi, id_match_statut, id_poule, id_equipe);
_set = (id_set , score , id_equipe, id_match);
utilise = (id_poule, id_terrain);
inscrit = (id_tournoi, id_equipe);
participe_au_set = (id_equipe, id_set);
participe_au_match = (id_equipe, id_match);

```

2 Langage de programmation procédurale

2.1 Première fonction

La première fonction que nous avons décidé de mettre en place permet de créer un événement et de lui attribuer un tournoi initial.

2.2 Seconde fonction

Cette fonction permet de d'inscrire une équipe dans un tournoi.
Elle est aussi capable de renvoyer une erreur en y précisant la raison.

2.3 Triggers

Le premier trigger permet d'affecter à une nouvelle poule un terrain disponible, on cherche un terrain qui n'est pas utilisé et en fonction de la disponibilité du terrain on lui affecte une poule.

La table match qui est liée à la table tournoi, représente une liaison de contexte par migration de clé id_tournoi vers la table match, pour illustrer qu'un match a été joué dans un tournoi. Le deuxième trigger s'applique sur ce contexte, on cherche les matchs qui ont été joués dont leur statut est FINI, une fois le nombre des matchs d'un tournoi en état FINI est égal au nombre des matchs liés à ce tournoi, on déclenche un trigger pour clôturer le tournoi c'est à dire basculer son statut vers 'CLOTURE'.

3 Tests

Les tests que nous avons décidé d'utiliser sur la base de données sont enregistrés dans le fichiers Test_groupeI.sql. Nous donnerons dans cette partie l'explication de nos test.

3.1 Tests fonction

Concernant la fonction nous voulions tester la fiabilité de la fonction et la gestion des erreurs. Dans le cas où on donne en paramètre à la fonction un tournoi qui n'existe pas dans la base de données nous obtenons le message d'erreur suivant : 'CE TOURNOI NEXISTE PAS'. Si on passe en paramètre une équipe qui n'est pas présente dans la base de donnée on nous renvoie : "CETTE EQUIPE NEXISTE PAS". Cette fonction reconnaît lorsqu'une équipe est déjà affectée, et rajoute dans la table tournoi les équipes qui y en été inscrites lorsque ça a été possible est affiche le message "INSCRIPTION A ETE EFFECTUEE AVEC SUCCES".

Pour la deuxième fonction, on vérifie que la valeur de retour de la fonction est une chaîne de caractères.

3.2 Tests Triggers

Pour le 1er trigger:

```
/* Insertion d'un terrain non disponible
INSERT INTO utilise VALUES('poule-3','12');

/* Affichage de l'erreur si la condition n'est pas vérifié
SELECT * FROM LOGERROR;

/* Affichage du terrain disponible qui a été affecté a une poule
SELECT * FROM utilise;
```

Pour le 2 ème trigger:

```
/* Mise a jour du statut du tournoi
UPDATE tournoi SET id-tournoi-statut="EN COURS";

/*Affichage de la table si condition est verifié
SELECT * FROM tournoi;

/* Affichage de l'erreur si la condition du trigger n'est pas vérifié
SELECT * FROM LOGERROR;
```

Conclusion

La réalisation de cette base de données nous a servi à mieux comprendre les notions vues en TP dans un autre SGBD différent de celui étudié en cours, ainsi que les mécanismes mis en place pour en faciliter l'utilisation, tel que les fonctions, et aussi les triggers.