

# UNIVERSITÉ DE MONTPELLIER

## L3 INFORMATIQUE

### TOURNOI SPORTIF

#### ”ESport”

RAPPORT DE PROJET

PROJET INFORMATIQUE COMMUN AVEC  
HLIN510 ET HLIN511

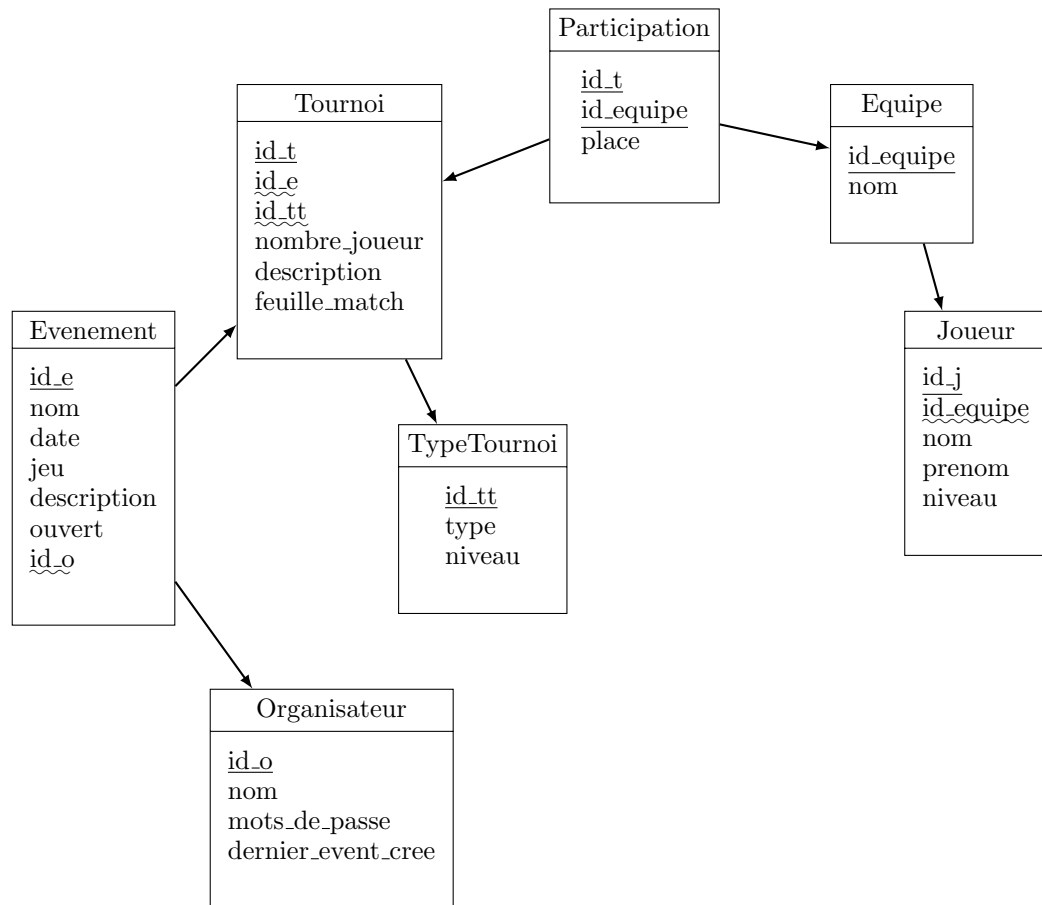
#### **GROUPE B**

M. Benjamin Baska  
Numero etudiant: 21503576  
M. Kevin Lastra  
Numero etudiant: 21706783

#### **Enseignant responsable:**

M. Pierre Pompidor  
M. Michele Meynard  
M. Pascale Poncelet

# 1 Modelé Entité/Association



Tables	Composant	Résumer
Evenement	id_e nom date jeu description ouvert id_o	Clé primaire Nom de l'événement Date de début de l'événement Jeu pour lequel est destinée l'événement Description de l'événement écrit par l'organisateur Boolean contrôlant l'ouverture et fermeture de l'événement Clé étrangère pour organisateur
Organisateur	id_o nom mot_de_passe dernier_event_cree	Clé primaire Nom d'utilisateur de l'organisateur Mot de passe de l'organisateur Dernière date à laquelle l'organisateur a créé un événement
Tournoi	id_t id_e id_tt nombre_joueur description feuille_match	Clé primaire Clé étrangère pour événement Clé étrangère pour TypeTournoi Nombre maximum d'équipe dans un tournoi Description du tournoi écrit par l'organisateur Format de text (JSON) pour sauvegarde les brackets
TypeTournoi	id_tt type niveau	Clé primaire Format de compétition des équipes, {1v1, 2v2, ...} Niveau du tournoi a titre indicatif
Participation	id_t id_equipe place	Clé primaire et Clé étrangère pour Tournoi Clé primaire et Clé étrangère pour Equipe Position final dans le tournoi
Equipe	id_equipe nom	Clé primaire Nom de l'équipe
Joueur	id_j id_equipe nom prenom niveau	Clé primaire Clé étranger pour équipe Nom du joueur Prénom du joueur Niveau du joueur indiquée par lui-même

## 2 Modelé Relationnel

evenement(id\_e, nom, date, jeu, description, ouvert, id\_o)

participation(id\_equipe,id\_t, place)

tournoi(id\_t, id\_tt, id\_e, nombre\_joueur, description, feuille\_match)

typetournoi(id\_tt, type, niveau)

equipe(id\_equipe, nom)

joueur(id\_j, id\_equipe, nom, prenom, niveau)

organisateur(id\_o , nom, mot\_de\_passe, dernier\_event\_cree)

## 3 Procédures Fonctionnels

Renvoi tout les évènement de l'organisateur passer en paramètre.

```
CREATE PROCEDURE event_par_org (IN org_id INT)
BEGIN
    SELECT e.id_e, e.nom
    FROM evenement e
    WHERE e.id_o = org_id;
END
```

Renvoi les stats d'un évènement passer en paramètre.

```
CREATE PROCEDURE stats_event(IN event_id INT)
BEGIN
    SELECT e.id_e, e.nom, e.id_o, o.nom, COUNT(t.id_t)
    FROM evenement e, tournoi t, organisateur o
    WHERE e.id_e = event_id AND e.id_o = o.id_o AND t.id_e = event_id
    GROUP BY e.id_e;

    SELECT t.id_t, COUNT(e.id_equipe)
    FROM equipe e, tournoi t, participation p
    WHERE t.id_e = event_id AND p.id_t = t.id_t AND p.id_equipe = e.id_equipe
    GROUP BY t.id_t;
END
```

## 4 Triggers

Quand un organisateur crée un évènement la date du dernier évènement crée est actualiser a la date actuelle.

```
CREATE TRIGGER update_organisateur_last_date AFTER INSERT ON evenement
FOR EACH ROW
BEGIN
    UPDATE organisateur
    SET dernier_event_cree = CURDATE()
    WHERE id_o = NEW.id_o;
END;
```

Quand un organisateur est supprimer, tout les évènement a son nom et changer pour l'organisateur para default "root".

```
CREATE TRIGGER delete_organisateur BEFORE DELETE ON organisateur
FOR EACH ROW
BEGIN
    UPDATE evenement SET id_o = 1
    WHERE id_o = OLD.id_o;
END;
```