BERNEY Alec, FORESTIER Quentin, HERZIG Melvyn

BDR – 19.01.2021

**Gestionnaire de tournois Rocket League**

Projet



Table des matières

[Introduction 2](#_Toc61814893)

[Cahier des charges 3](#_Toc61814894)

[Tournois 3](#_Toc61814895)

[Tours 4](#_Toc61814896)

[Série 4](#_Toc61814897)

[Matchs 4](#_Toc61814898)

[Equipes 4](#_Toc61814899)

[Joueurs 5](#_Toc61814900)

[Prix 5](#_Toc61814901)

[Utilisation 5](#_Toc61814902)

[Administrateur 5](#_Toc61814903)

[Utilisateur 5](#_Toc61814904)

[Modèle entité-association 7](#_Toc61814905)

[Choix effectués 8](#_Toc61814906)

[Modèle relationnel 10](#_Toc61814907)

[Contraintes d’intégrité 11](#_Toc61814908)

[Description des tables et attributs 12](#_Toc61814909)

[Manuel utilisateur 16](#_Toc61814910)

[Choix du mode 16](#_Toc61814911)

[Partie administrateur 17](#_Toc61814912)

[Formulaire principal de la partie administrative 17](#_Toc61814913)

[Formulaire Propriétés du tournoi 21](#_Toc61814914)

[Formulaire Ajout d’un tournoi 22](#_Toc61814915)

[Formulaire Edition des prix 22](#_Toc61814916)

[Formulaire Edition de la série 23](#_Toc61814917)

[Partie utilisateur 23](#_Toc61814918)

[Partie Commune 24](#_Toc61814919)

[Formulaire principale de la partie utilisateur 24](#_Toc61814920)

[Formulaire Statistiques 25](#_Toc61814921)

[Formulaire Tournoi 25](#_Toc61814922)

[Formulaire Série 26](#_Toc61814923)

[Responsable d’équipe 27](#_Toc61814924)

[Formulaire principale de la partie utilisateur 27](#_Toc61814925)

[Formulaire tournoi 28](#_Toc61814926)

[Formulaire Equipe 29](#_Toc61814927)

[Joueur faisant partie d’une équipe 31](#_Toc61814928)

[Formulaire Equipe 31](#_Toc61814929)

[Joueur sans équipe 31](#_Toc61814930)

[Formulaire Equipe 31](#_Toc61814931)

[Bugs connus 32](#_Toc61814932)

[Améliorations possibles 32](#_Toc61814933)

[Conclusion 34](#_Toc61814934)

[Annexes 35](#_Toc61814935)

[Guide d’installation 35](#_Toc61814936)

[Installer le programme 35](#_Toc61814937)

[Installation de la base de données 36](#_Toc61814938)

[Téléchargement de MySQL 36](#_Toc61814939)

[Configuration du serveur MySQL 37](#_Toc61814940)

[Installation de la base de données 40](#_Toc61814941)

[Fonctions stockées 43](#_Toc61814942)

[Procédures stockées 45](#_Toc61814943)

[Événements 48](#_Toc61814944)

[Application des contraintes d’intégrité 49](#_Toc61814945)

[Comportements mis en place 52](#_Toc61814946)

[Seeding 54](#_Toc61814947)

# Introduction

Ce projet est réalisé dans le cours de bases de données relationnels de la HEIG-VD. Il a pour but de mettre en pratique la conception et la réalisation d’une base de données sous MySQL.

Pour se faire, nous mettons en place une application pour gérer des tournois d’un jeu du nom de Rocket League. Le principe du jeu est proche du football.  Deux équipes, composées de trois joueurs, conduisant des [voitures](https://fr.wikipedia.org/wiki/Automobile), doivent frapper un ballon afin de marquer dans le but adverse.

Ce projet permettra aux utilisateurs de rejoindre des équipes pour participer à des tournois mis en place par des administrateurs.



Figure : <https://www.actugaming.net/rocket-league-1-million-joueurs-simultanes-361470/>

Les tournois adoptent le format suivant :



Figure Généré avec <https://challonge.com/fr>

# Cahier des charges

## Tournois

* Date de début et de fin
* Un nom
* Un nombre max d’équipe. Pour assurer que tout le monde commence au même tour
* Le nombre max est une puissance de 2 comprise entre 2 et 256.
* Un état :
  + « Attente » les équipes s’inscrivent et la date de début n’est pas encore atteinte,
  + « En cours » lorsque la date de début est dépassée
  + « Annulé » si la date de début est dépassée et qu’il manque des équipes. Suppression automatique après une semaine.
  + « Terminé » lorsque la finale a été disputée.
* Conserve quelle équipe a joué et avec qui au moment du tournoi.
* Un ou plusieurs tours.

## Tours

Lorsque les tournois sont créés, le nombre de tours nécessaire est déduit du nombre de participants.

Lors d’une rencontre entre deux équipes, plusieurs matchs sont disputés, c’est ce qu’on appelle une série. L’équipe qui peut passer au tour suivant est celle qui a remporté la majorité des matchs. En général une série s’étend sur 1-3-5-7 matchs.

* Nombre maximal de matchs de la série.
  + Indépendant entre les tours (p.ex. finale = 7 parties max, quart = 5 parties max)

## Série

* Deux équipes jouent une série
  + Le vainqueur est celui qui a remporté la majorité des matchs.

## Matchs

* Deux équipes
* Nombre de buts de chaque équipe
  + Buts par joueurs
  + Arrêts par joueurs
* Égalités impossibles

## Equipes

* 3 joueurs maximum
  + Dont un est responsable de l’équipe
  + Chaque joueur à une date d’arrivé et de départ.
* Un nom
* Un acronyme de trois caractères alphanumériques

-Une équipe qui n’a pas suffisamment de joueurs ne peut pas s’inscrire à un tournoi.

## Joueurs

* Nom
* Prénom
* Adresse mail
* Un pseudo
* Une date de naissance
* Une nationalité
* Est responsable d’une équipe au maximum ou aucune.

## Prix

* Un rang de l’équipe à attribuer (premier ou second uniquement).
* Est pécunier
  + Montant
* Est des objets
  + Nom

# Utilisation

Pour utiliser la plateforme, deux types de comptes sont proposés.

## Administrateur

* Créer des tournois, les modifier et les supprimer.
  + Définir les attributs du tournoi et par la suite changer la longueur des séries de chaque tour.
  + Changer les attributs du tournois et des tours tant qu’il n’a pas commencé., Corriger/entrer les résultats d’un match une fois qu’il est en cours. Il est possible de retirer une équipe inscrite tant que le tournoi est en attente.
  + Supprimer ne conserve aucune trace du tournoi.
* Voir les tournois par états.
* Connaître la moyenne de la vitesse d’inscription pour les tournois en fonction du nombre d’équipe.
* Connaître la moyenne du nombre d’équipe par joueur.

## Utilisateur

* Création d’équipe, il en devient responsable.
  + Accepter des joueurs, ils seront affichés avec leur total et moyenne de but et d’arrêt.
  + Retirer des joueurs.
  + Inscription à un tournoi, si équipe complète.
  + Retirer du tournoi, si « En attente » sinon abandonner et perdre la série instantanément
  + Afficher les performances de ses joueurs (p.ex. nombre de séries gagnées par joueur)
  + Le responsable ne quitte pas son équipe et ne la supprime pas non plus. Cette fonctionnalité est laissée pour une version future du projet.
* Rejoindre une équipe non pleine.
* Quitter une équipe que si l’équipe n’a que des tournois « Annulés » ou « Terminé »
* Consulter ses performances :
  + Total et moyenne de but et d’arrêt.
  + #séries gagnées / #séries perdues.
* Voir un historique de ses participations aux tournois et avec qui (équipe, coéquipiers).

# Modèle entité-association



# Choix effectués

**Prix**

Pour l’attribution des prix, nous avons décidé de ne pas vérifier si le prix du premier est plus important que le prix du deuxième. Comme des objets sont pris en compte et que nous ne stockons pas leur valeur, nous ne pouvons donc pas déterminer si l’importance des prix, en fonction du rang des équipes, est respectée. Cette tâche revient donc à l’administrateur du tournoi.

**Tournoi**

L’état du tournoi n’est pas stocké en tant que telle dans la base de données car il peut être déduit en fonction d’autres attributs/enregistrements (p.ex. la date de début du tournoi).

Nous avons uniquement autorisé un nombre d’équipe maximal par tournoi égal à une puissance de deux comprise entre 2 et 256 pour permettre à chaque équipe de discuter le même nombre de rencontres et ainsi avantager personne.

La date de fin du tournoi ne sert pas à indiquer quand le tournoi doit se terminer mais quand il s’est terminé. De ce fait, à la création du tournoi, dateFin sera nulle et sera automatiquement mise à la date courante quand la finale sera terminée.

Nous n’avons pas de précision quant à la cardinalité entre tournoi et prix. Il est donc possible d’avoir un tournoi sans prix et un tournoi qui contient plusieurs prix.

**Tour**

Les tours sont numérotés de manière décroissante où la finale sera toujours le tour d’identifiant 1.  
Comme il y a au maximum 256 équipes dans un tournoi et au minimum 2, nous obtenons une cardinalité 1 à 8 entre tournoi et tour.

Pour le premier tour du tournoi les équipes sont traitées dans leur ordre d’inscription. P.ex. la première inscrite jouera contre la seconde, la troisième contre la quatrième, etc…

**Série**

Comme un tour contient entre 2 et 256 équipes, il y a entre 1 et 128 séries disputées par tour.

Le gagnant d’une série n’est pas explicitement stocké, il est déduit en fonction de l’équipe qui a remporté le plus de matchs associés à la série.

**Match**

Comme une série se joue en 1,3,5,7 matchs et qu’elle s’arrête dès qu’une équipe a remporté la majorité des matchs, on obtient une cardinalité 2 à 7 entre match et série.

Nous ne stockons pas les buts des équipes dans un match car ils peuvent être calculé en fonction de leurs joueurs entre match et joueur.

Le gagnant d’un match n’est pas explicitement stocké. Il est déduit en fonction du résultat du match, c’est-à-dire l’équipe qui comptabilise le plus de goal.

**Match Joueur**

Si un joueur ne marque pas de buts et n’arrête aucun but, il a tout de même un enregistrement. Avec 0 dans les attributs correspondants

**Equipe**

Les équipes ne sont pas uniquement liées à un tournoi. Elles peuvent donc s’inscrire à plusieurs tournois consécutivement.

**Serie Equipe**

Nous avons conservé ce lien bien qu’il crée une boucle entre Serie, Match Joueur et Equipe afin de pouvoir vérifier plus facilement si la participation d’un joueur à un match est légale.

**Joueur**

La date de naissance du joueur n’a pas de limite d’âge minimal pour utiliser la plateforme. De ce fait, nous acceptons des utilisateurs de tout âge.

**Joueur Equipe**

Comme les champs de date d’arrivé et de départ correspondent à la date d’entrée et de départ dans l’équipe, ils seront automatiquement initialisés à la date du jour, respectivement quand le responsable accepte le joueur et que le joueur quitte l’équipe.

Cette table contient la date d’arrivée dans une équipe comme composante de la clé primaire afin de pouvoir permettre à un joueur de rejoindre une équipe dans laquelle il a déjà joué.

Un joueur qui postule dans une équipe créer un enregistrement avec date arrivée 0000 :00 :00

Dès qu’une équipe est rejointe, l’enregistrement associé est mis à jour et toutes ses demandes restantes sont supprimées

**Prix**

Nous avons séparé les prix sous forme d’objet des prix sous forme d’argent. Comme seuls, les premiers et les seconds peuvent remporter des prix, nous avons intégré les montants pécuniers directement dans le tournoi. Ces montants peuvent être nuls. Cependant les objets ont été modélisés à part pour pouvoir être réutilisé entre les tournois.

# Modèle relationnel

# Contraintes d’intégrité

**Tournoi**

* Le nombre d’équipe d’un tournoi = 2^x ou x <- [1, 2 .. 8].
* dateHeureDebut est strictement plus petite que dateHeureFin.
* dateHeureDebut est strictement plus grande que la date courante.

**Tournoi\_Equipe**

* La date d’inscription d’une équipe est plus vieille que la dateDebut du tournoi.

**Tour**

* Le champ longueurMaxSerie = [1,3,5,7].

**Serie**

* Référence deux équipes inscrites au tournoi associé et non éliminées.
* Une équipe ne peut pas gagner si elle joue contre aucune autre.

**Match**

* Le nombre de buts total des deux équipes ne peuvent pas être égaux.

**Equipe**

* Ne peut pas s’inscrire à un tournoi si elle n’a pas trois joueurs.
* Ne peut pas s’inscrire à un nouveau tournoi tant qu’elle est inscrite à un tournoi en « attente » ou « en cours ».
* Peut seulement s’inscrire à un tournoi « en attente ».
* Le responsable est forcément un joueur

**Joueur**

* Ne peut pas quitter son équipe tant qu’elle participe à un tournoi.
* Un joueur ne joue que dans une équipe à la fois.
* La date de naissance doit être plus petite que la date courante.
* L’email doit être unique

**Equipe\_Joueur**

* Le champ dateHeureArrivee est forcément plus petite que dateHeureDepart.
* Un enregistrement ne peut être créé pour un joueur tant qu’il est déjà dans une équipe.

**Match\_Joueur**

* Les buts et les arrêts sont positifs ou nuls.
* Un joueur doit avoir des buts/arrêts (minimum 0) que dans un match auquel il a participé.
* Référence un joueur qui joue pour une des deux équipes de la série.

**Objet**

* Le nom doit être unique

**Prix**

* Positif ou nul

# Description des tables et attributs

|  |
| --- |
| Objet  Définit les objets qui peuvent être ajouté au prix, en plus de l’argent. |
| id Identifiant de l’objet. |
| nom Nom de l’objet. |

|  |
| --- |
| Prix\_Objet Lie un prix à un/des objet(s). |
| idPrix Identifiant du prix qui inclut l’objet. |
| idObjet Identifiant de l’object inclus. |

|  |
| --- |
| Prix Définit un prix pécunier pour un tournoi. |
| id Identifiant du prix. |
| montantArgent Somme à remporter. |

|  |
| --- |
| Tournoi Définit un tournoi |
| id Identifiant du tournoi. |
| dateHeureDebut Date et heure où le tournoi débute. |
| dateHeureFin  Date et heure où le tournoi s’est terminé.  NULL tant que le tournois est « en cours » ou « en attente ».  Egal à dateHeureDebut si « Annulé ». Différent de dateHeureDebut quand « Terminé ». |
| nom Nom du tournoi. |
| nbEquipesMax  Nombre d’équipes maximal et nécessaire au tournoi. [2,4,8,16,32,64,128,256] |
| idPrixPremier Prix remporté par l’équipe gagnante. |
| idPrixSecond Prix remporté par l’équipe seconde. |

|  |
| --- |
| Tournoi\_Equipe Définit les inscriptions pour les tournois. |
| idTournoi Numéro du tournoi de l’inscription. |
| acronymeEquipe Acronyme de l’équipe inscrite. |
| dateInscription Date où l’inscription a eu lieu. |

|  |
| --- |
| Tour Définit les tours dans un tournoi. |
| no Numéro du tour. (1 étant le tour de la finale). |
| longueurMaxSerie Nombre maximum de matchs jouables dans les séries du tour. |
| idTournoi Identifiant du tournoi d’attache. |

|  |
| --- |
| Serie  Définit une série entre dans équipe dans un tour. |
| id  Identifiant de la série. (De 1 à noTour) |
| noTour  Numéro du tour d’attache. |
| idTournoi  Identifiant du tournoi dans lequel est le tour. |
| acronymeEquipe1  Acronyme de la première équipe qui dispute la série. |
| acronymeEquipe2 Acronyme de la seconde équipe qui dispute la série. |

|  |
| --- |
| Match Définit un match dans une série. |
| id Identifiant du match (de 1 à longueurMaxSerie). |
| idSerie Identifiant de la série d’attache. |
| noTour Numéro du tour de la série. |
| idTournoi Numéro du tournoi du tour. |

|  |
| --- |
| Match\_Joueur Défini les performances d’un joueur dans un match. |
| idJoueur  Identifiant du joueur. |
| nbButs  Nombre de buts marqués. |
| nbArrets Nombre d’arrêts effectués. |
| idMatch Identifiant du match associé. |
| idSerie Identifiant de la série du match. |
| noTour Numéro du tour de la série. |
| idTournoi Identifiant du tournoi associé au tour. |

|  |
| --- |
| Joueur Définit un utilisateur du système. Un utilisateur est joueur. |
| id Identifiant de l’utilisateur. |
| nom Nom du joueur |
| prenom Prénom du joueur. |
| email Adresse de messagerie. |
| pseudo  Pseudo à utiliser sur la plateforme. |
| dateNaissance  Date de naissance de la personne. |

|  |
| --- |
| Equipe Définit une équipe qui peut participer aux tournois. |
| acronyme Identifiant unique de l’équipe. (Entre 1 et 3 caractère) |
| nom  Nom complet de l’équipe. |
| idResponsable  Identifiant du joueur qui a créé l’équipe. |

|  |
| --- |
| Equipe\_Joueur Définit les joueurs au sein d’une équipe, passés ou présents. |
| acronymeEquipe Identifiant de l’équipe. |
| idJoueur Identifiant du joueur référencé |
| dateHeureArrivée  Date et heure d’arrivée du joueur dans l’équipe.  Si ‘0001-01-01 00 :00 :00’ la demande d’adhésion est en cours  sinon le joueur est/a été accepté dans l’équipe. |
| dateHeureDepart  Date et heure du départ du joueur de l’équipe.  Si NULL le joueur n’a pas encore quitté l’équipe  Sinon il a quitté l’équipe. |

# Manuel utilisateur

## Choix du mode

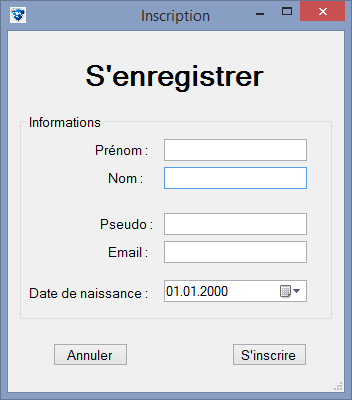
Notre application, contient deux modes :

Le mode administrateur accessible via le bouton « Administration » qui sert à gérer tous les aspects de création, modification et suppressions des tournois.

Le mode utilisateur accessible en entrant l’email d’un utilisateur lambda, un joueur dans le champ texte.

Une fois l’email entré, il suffit de cliquer sur le bouton «  Se connecter » pour accéder à la partie utilisateur du programme.

Il également possible de créer un nouveau compte / joueur en cliquant sur le bouton « S’inscrire ».

Dans le cas de l’inscription, un nouveau formulaire s’ouvre où tous les champs doivent être minutieusement rempli afin de créer un nouveau joueur.

Une fois tous les champs remplis, il suffit de cliquer sur le bouton « S’inscrire » pour finaliser l’inscription.

Le bouton « Annuler » quitte le formulaire.

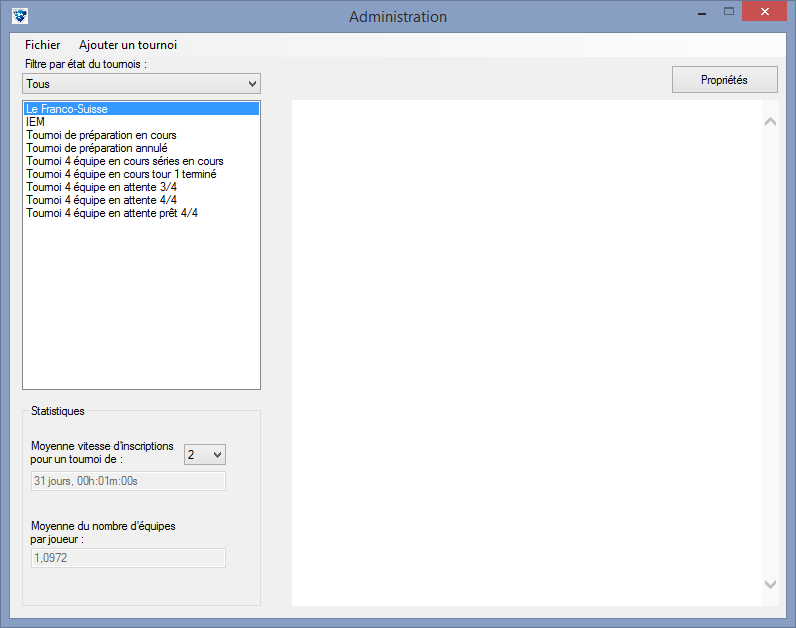
## Partie administrateur

### Formulaire principal de la partie administrative

Voici l’état de base en arrivant dans la partie administration de notre gestionnaire de tournoi.

Tout en haut, nous avons un menu possédant les possibilités « Fichier » et « Ajouter un tournoi ».

En cliquant sur « Fichier » puis « Choix du mode », on peut revenir au menu principal. Si l’on clique sur « Ajouter un tournoi », un nouveau formulaire apparaît.

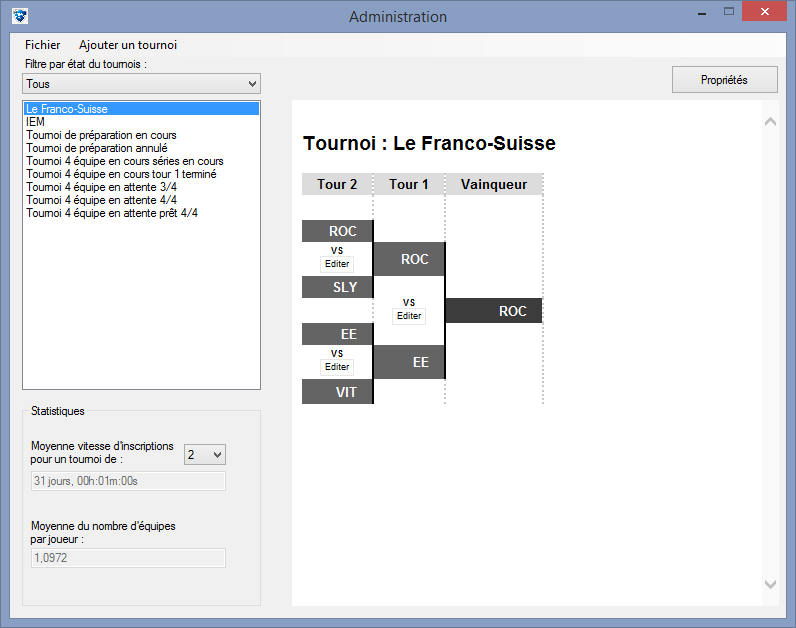


Nous sautons pour l’instant la liste déroulante pour parler de la liste de tournois affichés.

La liste contient tous les noms des tournois affichés et pouvant être sélectionné. Il suffit de sélectionner un nom de tournoi (ici surligné en bleu) pour afficher son arbre de tournoi dans la fenêtre à droite.

Il y a également un bouton « Propriétés », tout en haut à droit, permettant d’afficher les propriétés du tournois sélectionnés dans la liste. Ce dernier ouvrira un nouveau formulaire d’afficher et modifier les propriétés du tournoi.

On peut également interagir avec l’arbre de tournoi qui possède un bouton « Editer » pour chaque série. Ce bouton permet d’afficher et de modifier les résultats de la série dans un nouveau formulaire.



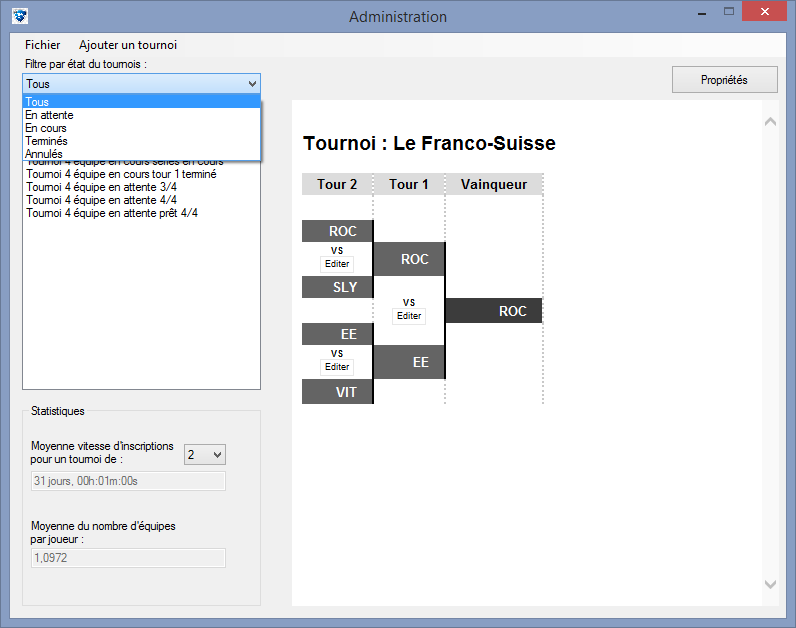
Les formulaires des propriétés du tournoi et des séries seront présentés plus loin dans le document.

Revenons à notre liste déroulante située un peu plus bas par rapport au menu. Cette dernière permet d’afficher seulement les tournois qui sont dans l’état choisi par l’utilisateur.

Les possibilités sont les suivants :

* Tous
* En attente
* En cours
* Terminés
* Annulés

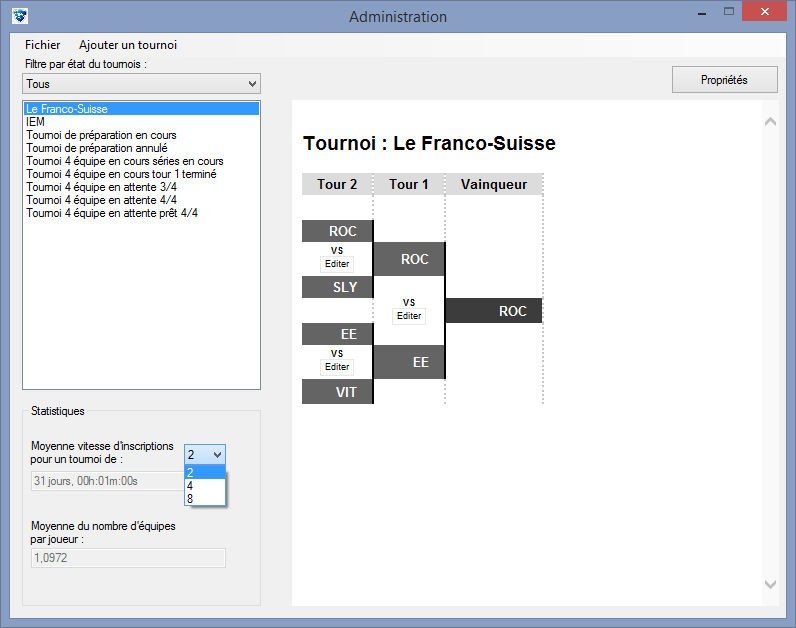
En sélectionnant un choix, tous les tournois correspondant à la sélection seront affichés dans la liste en dessous.



Tout en bas, dans la partie « Statistiques », on peut remarquer un liste déroulante servant à afficher la vitesse moyenne d’inscriptions pour un tournoi du choix sélectionné dans cette dernière.

Cette liste déroulante est remplie en fonction des tournois créés à ce jour. En effet, c’est le nombre d’équipes pouvant participer à tournoi qui est affiché dans la liste déroulante. Si tous les tournois créés n’acceptent que 4 joueurs, la liste déroulante contiendra uniquement la valeur 4.

Nous avons également la moyenne du nombre d’équipe par joueurs représentée dans un champ en dessous. Cette statistique est indépendante de la liste déroulante.



### Formulaire Propriétés du tournoi

Revenons maintenant au formulaire permettant de modifier les propriétés du tournoi sélectionné dans le formulaire principal.

Dans la partie « Informations sur le tournoi », nous avons d’abord les informations du tournoi sélectionné qui sont affichées dans les champs correspondants.

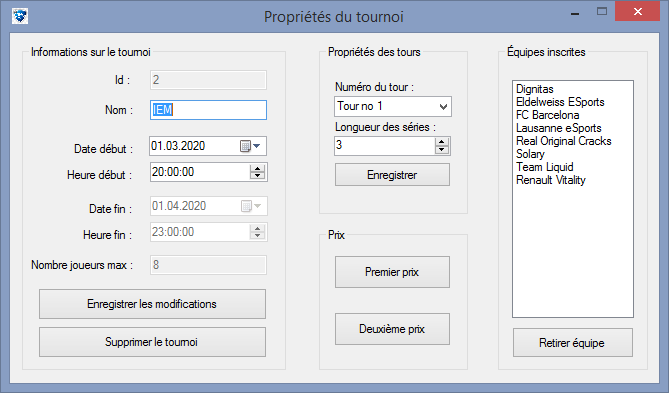
Pour les modifier, il suffit de clique sur le champ voulu et modifier sa valeur. Il faut ensuite clique sur « Enregistrer les modifications » pour finaliser la chose.

Le bouton « Supprimer le tournoi » permet de supprimer le tournoi définitivement.

Dans la partie « Propriétés des tous », on peut modifier La longueur des séries du tour voulu à l’aide d’une liste déroulante et d’un champ de valeurs définies (nombre impaire). Une fois les valeurs modifiées, il suffit d’appuyer sur « Enregistrer » pour finaliserle tout.

Les boutons « Premier prix » et « Deuxième prix » de la section « Prix » ouvrent un nouveau formulaire permettant de modifier le prix du premier ou du second.

Il est également possible de retirer une équipe du tournoi dans la partie «Equipes inscrites ». Pour réaliser ceci, il suffit de sélectionner une équipe participante dans le liste et d’ensuite cliquer sur le bouton « Retirer équipe » pour finaliser la chose.



### Formulaire Ajout d’un tournoi

Voici le petit formulaire permettant de créer un tournoi.

Il suffit de remplir tous les champs du formulaire et de cliquer sur « Ajouter » pour créer un nouveau tournoi.

### Formulaire Edition des prix

Commençons par la partie « Prix assignés ». Cette partie contient le prix associé au premier ou deuxième en fonction du bouton pressé dans le formulaire « création de tournoi ».

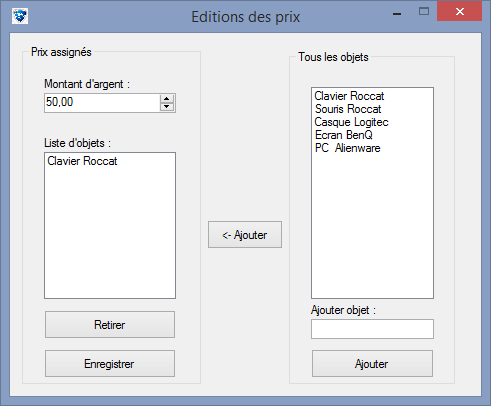
Chaque tournoi a forcément un montant d’argent indiqué dans son propre champ. Il est possible de le modifier en changeant tout simplement la valeur de son champ.

On peut retirer un objet en le sélectionnant dans la liste et en cliquant ensuite sur le bouton « Retirer ».

Dans la partie « Tous les objets », on peut ajouter un objet de notre choix en remplissant le champ destiné à ceci et en cliquant ensuite sur le bouton « Ajouter ». Ceci ajoutera l’objet dans la liste des objets possibles pour tous les tournois.

Il est également possible d’ajouter un objet en tant que prix pour le tournoi en le sélectionnant dans la liste et en appuyant ensuite sur le bouton « Ajouter ».

On peut enregistrer toutes les modifications des prix du tournoi et son montant d’argent en cliquant sur le bouton « Enregistrer ».

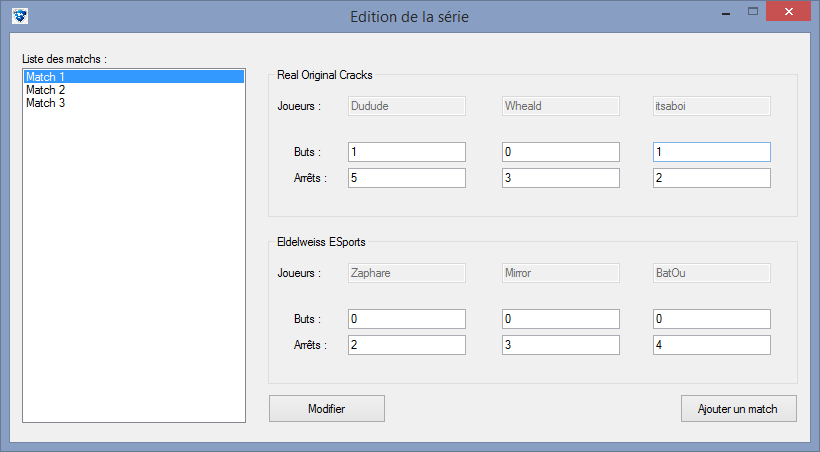


### Formulaire Edition de la série

Ce formulaire sert à remplir ou modifier les informations d’une série entre 2 équipes.

À gauche, nous avons la liste des matchs d’une série. On peut sélectionner un match (ici surligné en bleu foncé) pour afficher ses informations. Il est ensuite de les modifier en changeant les informations dans les champs voulus et ensuite appuyer sur le bouton « Modifier ».

Il est également possible d’ajouter un match à la série en cliquant sur le bouton « Ajouter un match ».



## Partie utilisateur

Cette partie utilisateur est divisée en plusieurs sections. En effet, en fonction du statut du joueur (Responsable, avec équipe, sans équipe) les formulaires changent.

Alors pour illustrer toutes les possibilités, nous allons utiliser les utilisateurs suivants :

* Responsable équipe pleine : quentin.forestier@heig-vd.ch
* Responsable équipe non pleine avec une demande : johnny@uc1.com
* Responsable équipe avec anciens joueurs : Jujuju@lse.ch
* Joueur sans équipe : gabin@sansequipe.com
* Joueur lambda avec un équipe : melvyn.herzig@heig-vd.ch

Nous allons donc d’abord expliquer comment utiliser la partie commune à toutes les sections, puis expliquer chaque spécificité de chaque section.

## Partie Commune

Elle regroupe toutes les fonctionnalités et affichages communs.

### Formulaire principale de la partie utilisateur

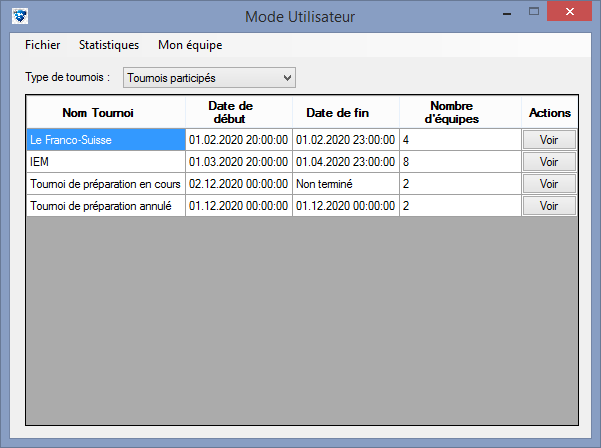
Voici l’état de base en arrivant dans la partie utilisateur de notre gestionnaire de tournoi.

Tout en haut, nous avons un menu possédant les possibilités « Fichier », « Statistiques » et « Mon équipe ». Voici ce qui va se passer en cliquant sur :

* « Fichier » puis « Choix du mode », on peut revenir au menu principal.
* « Statistiques », un nouveau formulaire contenant les statistiques du joueur apparaît.
* « Mon équipe », un nouveau formulaire contenant des informations sur l’équipe du joueur apparaît.

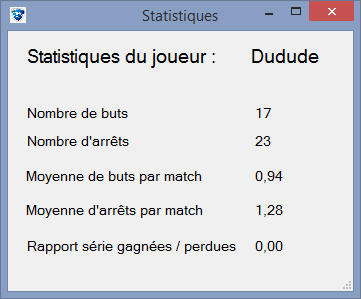
On peut également cliquer sur le bouton « Voir » pour afficher le tournoi de la ligne du bouton sélectionné dans un nouveau formulaire.

La liste déroulante en haut apparaît uniquement si le joueur est responsable d’équipe.



### Formulaire Statistiques

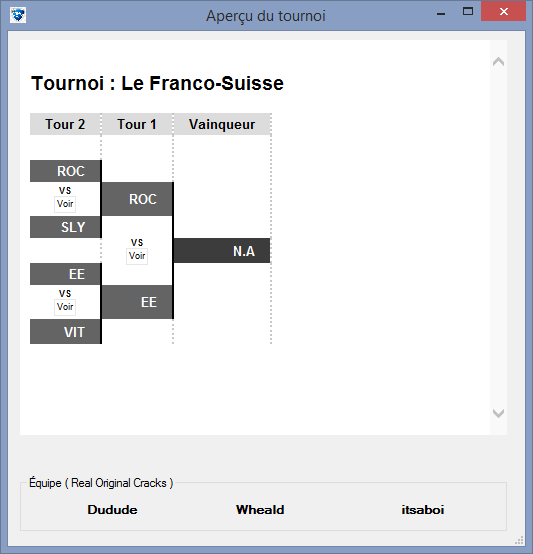
Ce formulaire ne contient aucune action possible. Il est juste présent pour afficher les statistiques du joueur connecté.



### Formulaire Tournoi

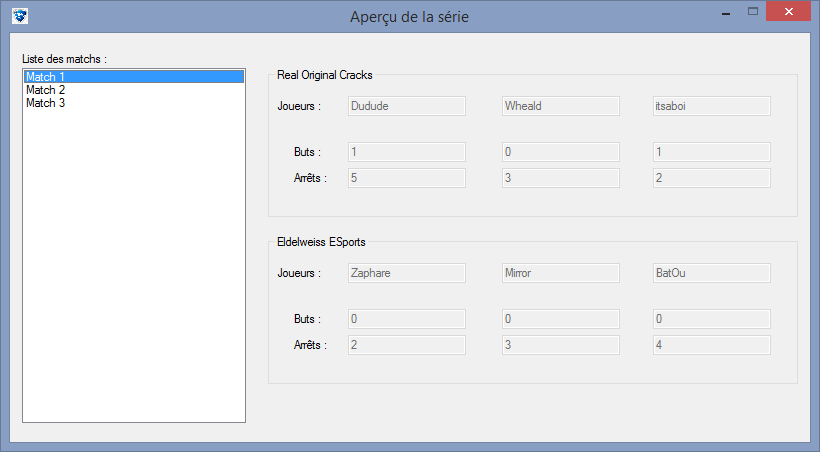
Ce formulaire affiche l’arbre de tournoi sélectionné dans le formulaire principal. On peut également y trouver les joueurs de l’équipe avec laquelle il a participé au tournoi.

La seule action possible est de cliquer sur un des boutons « Voir » pour afficher un nouveau formulaire contenant les informations détaillées de la série sélectionnée.



### Formulaire Série

Ce formulaire a pour but d’afficher tous les résultats de la série sélectionnée dans le formulaire Tournoi. La seule action possible et de sélectionné dans la liste des matchs à gauche, le match dont l’on veut voir les informations détaillées à droites.



## Responsable d’équipe

### Formulaire principale de la partie utilisateur

La seule différence avec la partie commune est que l’on peut utiliser la liste déroulante contenant les possibilités :

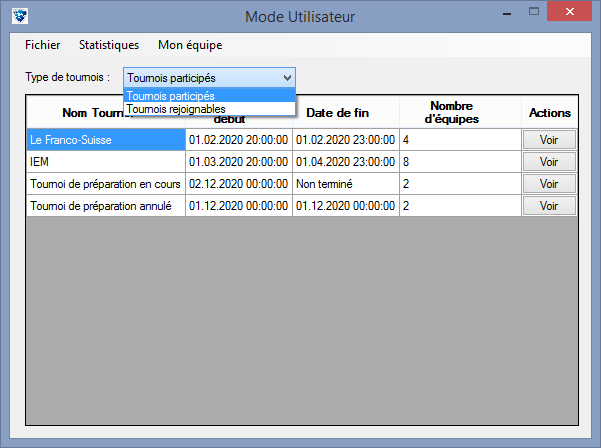
* Tournois participés
* Tournois rejoignables

Les tournois affichés dans le tableau en dessous dépendent du choix de la liste déroulante.

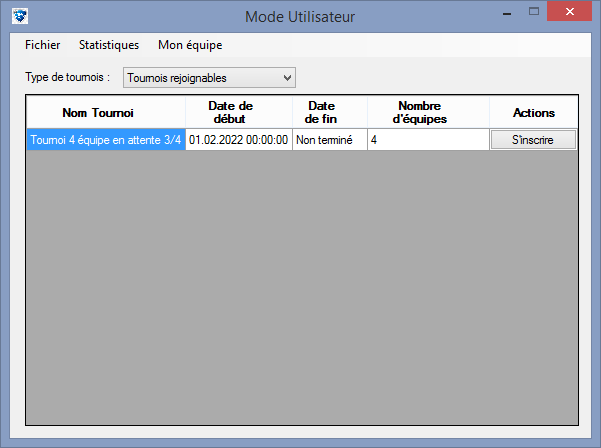
Nous avons également un bouton cliquable qui change de valeur et de fonctionnalité en fonction du choix des tournois affichés. Le bouton peut prendre les valeurs :

* « Voir » pour voir l’arbre du tournoi sélectionné (formulaire Tournoi)
* « S’inscrire » pour s’inscrire au tournoi sélectionné

Nous avons ici l’affichage des tournois auxquels nous avons participés et qui sont donc fini :

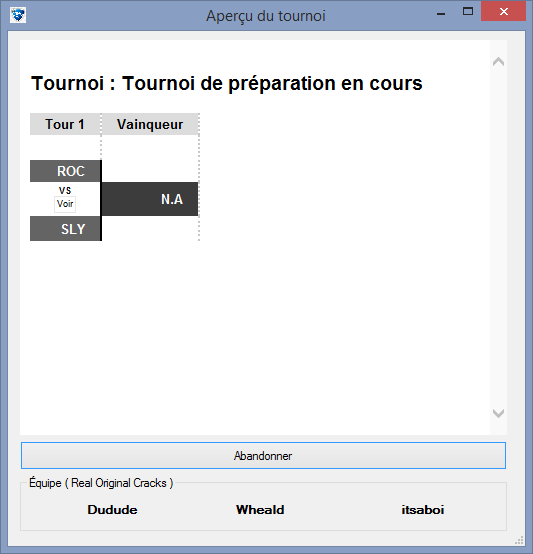


Voici l’affichage des tournois rejoignables :



### Formulaire tournoi

Quand on sélectionne un tournoi « Non terminé » dans le formulaire principal et qu’on est responsable d’une équipe, il est possible d’abandonner le tournoi en cliquant sur le bouton « Abandonner ».



### Formulaire Equipe

Le but de ce formulaire au niveau responsable est de gérer son équipe au complet.

La première fonctionnalité est une liste déroulante contenant les possibilités :

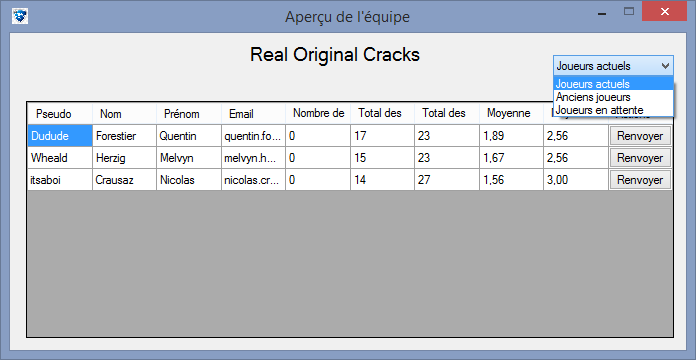
* Joueurs actuels
* Anciens joueurs
* Joueurs en attente

La liste déroulante, en fonction du choix effectué parmi les possibilités énoncées, modifie les joueurs affichés dans le tableau en dessous. Les données des joueurs sont leurs informations personnelles ainsi que leurs statistiques de jeu.

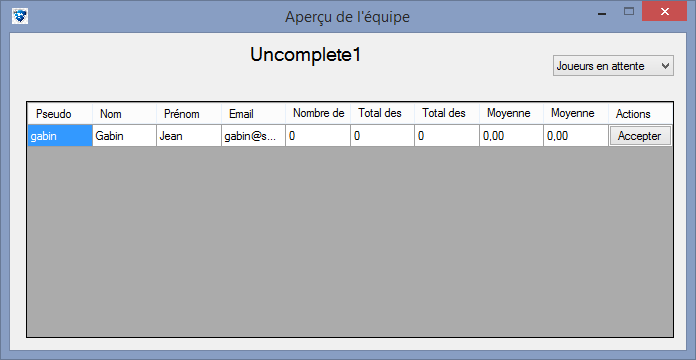
Nous avons également un bouton cliquable qui change de valeur et de fonctionnalité en fonction du choix des joueurs affichés. Le bouton peut prendre les valeurs :

* « Renvoyer » pour voir l’arbre du tournoi sélectionné (formulaire Tournoi)
* « Accepter » pour quitter un tournoi rejoins au préalable et non terminé

Voici la première partie du formulaire :



Voici le formulaire si on veut afficher les joueurs en attente :



Voici le formulaire si on veut afficher les anciens joueurs :



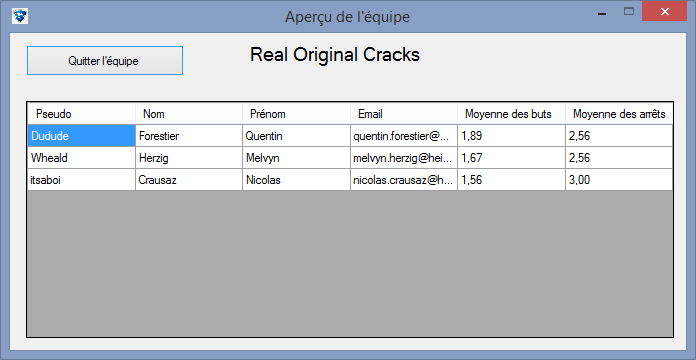
## Joueur faisant partie d’une équipe

### Formulaire Equipe

Le but de ce formulaire au niveau joueur ayant déjà une équipe est de pouvoir quitter cette dernière et afficher ses coéquipiers.

Comme dit plus haut, les coéquipiers sont affichés dans le tableau avec leur informations personnelles et quelques statistiques de jeu.

La seule action possible est de quitter définitivement l’équipe en cliquant sur le bouton « Quitter l’équipe ».



## Joueur sans équipe

### Formulaire Equipe

Le but de ce formulaire au niveau joueur n’ayant pas d’équipe est d’en rechercher une ou d’en créer une.

Pour réaliser tout ceci, une liste des équipes rejoignables.

Pour réaliser une demande d’adhésion à une équipe, il suffit d’en sélectionner une dans la liste et de cliquer sur le bouton « Rejoindre ».

Pour créer une équipe, il suffit de remplir minutieusement les champs « Acronyme » et « Nom complet » et d’ensuite clique sur le bouton « Créer » pour finaliser la création d’une nouvelle équipe dont le joueur actuel sera responsable.

# Bugs connus

* Possibilité de modifier les scores de la finale, même si celle-ci est terminée.
* Possibilité de donner des noms comme «  » ou « . » aux tournoi, joueurs, équipes et objets.
* Possibilité d’abandonner une série seulement lorsque l’adversaire est connu.
* Ajouter deux fois le même objet ne fonctionne pas.
* Si une équipe, avec des joueurs qui ont leurs statistiques vierges à 0, gagnent une série par forfait, les buts, buts/match passent à 1 alors qu’ils n’ont techniquement pas joué le match. La source du bug est connue, pour abandonner une série, nous complétons les matchs restants en mettant 1 but à chaque joueur de l’équipe gagnant et 0 aux perdant.
* Les tournois qui sont annulés n’apparaissent pas immédiatement dans la catégorie de ce nom. Ce comportement est causé par l’événement qui détecte les tournois annulés une fois par heure. Ce comportement est connu et voulu mais peut surprendre et être considéré comme un bug.
* Les matchs vérifient s’ils sont égalités lorsqu’une entrée Match\_Joueur est insérée/mise à jour et que le match en comptabilise 6. Quand nous mettons à jour un match, nous mettons à jour séquentiellement les entrées des 6 joueurs. Bien qu’un match ne soit pas égalité en prenant en compte les 6 joueurs, la séquence des mises à jour peut entraîner des états intermédiaire égalité, ce qui empêche la modification du match.
* Quand un joueur quitte son équipe et qu’il y retourne, il apparait dans les « Joueurs actuel » mais également toujours dans les « Anciens joueurs ».
* La fonction d’abandon d’une série peut entraîner un comportement indéfini si deux équipes de la même série décident d’abandonner en même temps.

# Améliorations possibles

Bien que fonctionnel, notre système peut encore être amélioré pour une utilisation plus agréable tant d’un point de vue utilisateur que développeur.

**Application**

* Ajouter des quantités aux objets lorsqu’ils sont attribués à un prix.
* Permettre la participation à plusieurs tournois qui ne se chevauchent pas. En l’état une équipe doit avoir terminé tous ses tournois avant de s’inscrire à un nouveau.
* Meilleure gestion des équipes. Le responsable devrait pouvoir quitter son équipe, nommer un remplaçant ou même supprimer son équipe s’il le souhaite.
* Permettre aux tournois de démarrer même si le nombre maximal d’équipe n’est pas atteint.
* Ajouter des champs textuels de recherches pour les tournois et les noms d’équipes.

**Base de données**

Le modèle adopté est très strict au niveau des contraintes d’intégrité. Pour éviter de mauvaises manipulations en insertion/mise à jour/suppression nous avons créé des vérifications dans des triggers. Il est souvent difficile de suivre ce qui se passe et à quel moment intervient quelle vérification. Un remaniement du schéma pourrait être à envisager car même l’utilisateur root est « limité dans ses actions ».

Pour limiter les droits de l’application nous avons utilisé un unique utilisateur « GTRL\_Manager » qui a le plein pouvoir sur la base de données de l’application. Une solution au problème précédent aurait été d’affiner les droits de cet utilisateur sur certaines procédures/tables. Ainsi seules un certain nombre d’actions prédéterminées auraient besoin d’être vérifiées. En conséquence moins de vérifications serait nécessaires, le code serait donc plus lisible et l’utilisateur root retrouvait le plein pouvoir.

Ainsi un ajustement du modèle de la base de données ou des droits utilisateurs permettraient de rendre le code plus lisible et plus souple.

La fonction d’abandonner est codée dans le code de l’application. La manière correcte de l’effectuer aurait été de l’implémenter dans la base de données avec une transaction. Malheureusement nous n’avons pas réussi à l’implémenter convenablement, nous l’avons donc temporairement développée dans l’application.

# Conclusion

Au terme de ce projet, nous sommes parvenus concevoir notre base de données ainsi qu’une application permettant de l’utiliser.

Nous avons réussi à respecter l’ensemble de notre cahier des charges. Toutefois, notre système n’est pas exempt de bugs. Certains auraient pu être éviter avec un modèle différent, d’autres avec un peu plus d’attention de notre part.

Nous sommes satisfaits du travail rendu, bien qu’il nous ait demandé plus de temps que le temps prévu pour le mettre en place.

# Annexes

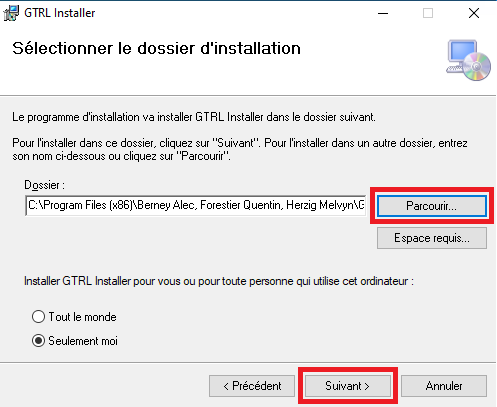
## Guide d’installation

### Installer le programme

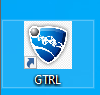
Afin d’installer le programme, lancer « GTRL\_Installer.msi ». Ceci installera le gestionnaire de tournoi, les librairies ainsi que les ressources nécessaires au fonctionnement du programme.

Il est possible que votre anti-virus détecte l’installateur comme dangereux. Si c’est le cas, désactiver votre anti-virus pendant l’installation ou autoriser le programme et l’installeur dans ce dernier.

Une fois l’installeur lancé, il est d’abord écrit un message de bienvenue dans l’assistant d’installation du programme, cliquer sur « Suivant ». Il est ensuite demandé « où placer le programme sur l’ordinateur ». Indiquer l’emplacement souhaité en utilisant le bouton « Parcourir » puis cliquer sur « Suivant » et encore une fois « Suivant », puis finalement « Fermer ».



Après quelques chargements, une fenêtre indique que l’installation est terminée. Un raccourci pour lancer l’application a été créé sur le bureau.



/!\ Souvenez-vous où le programme a été installé. Il faut aller chercher ultérieurement un fichier à cet emplacement.

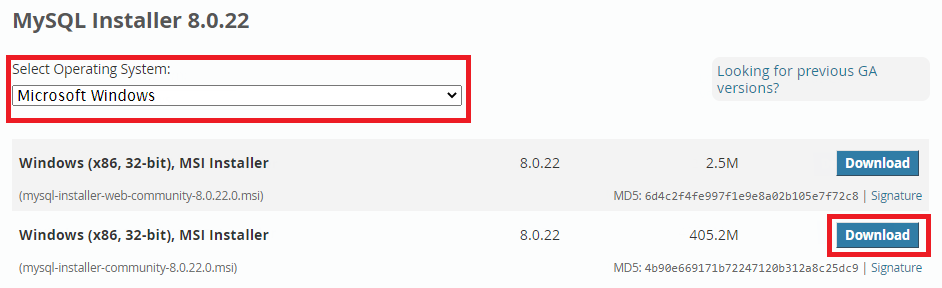
## Installation de la base de données

### Téléchargement de MySQL

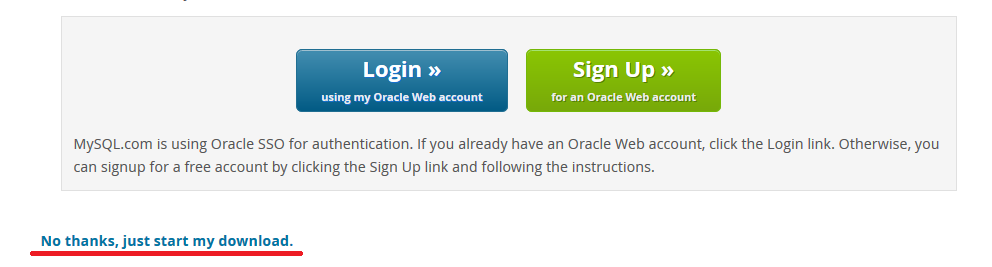
Afin de pouvoir utiliser l’application, l’installation manuelle de la base de données et d’un SGBD (système de gestion de bases de données) sont nécessaires.

Pour ceci, se rendre sur : <https://dev.mysql.com/downloads/installer/> afin de télécharger le système de gestion de bases de données MySQL.

Choisir « Windows » en guise de système d’exploitation puis cliquer sur le bouton « Download » de l’installeur non web.



Ceci nous dirige sur une page de téléchargement. Il n’est pas nécessaire de s’inscrire. Il suffit de cliquer sur « No thanks, just start my download » en dessous des boutons « Login » et « Sign up ».



Après le téléchargement, nous obtenons donc :

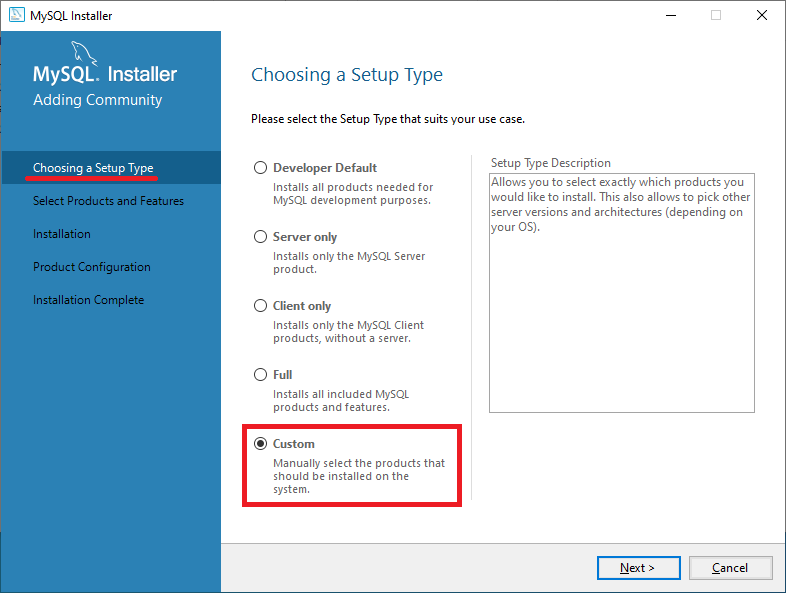


Lancer l’installation en double cliquant sur l’installeur. Autoriser l’application à apporter des modifications à votre appareil en appuyant sur « Oui ».

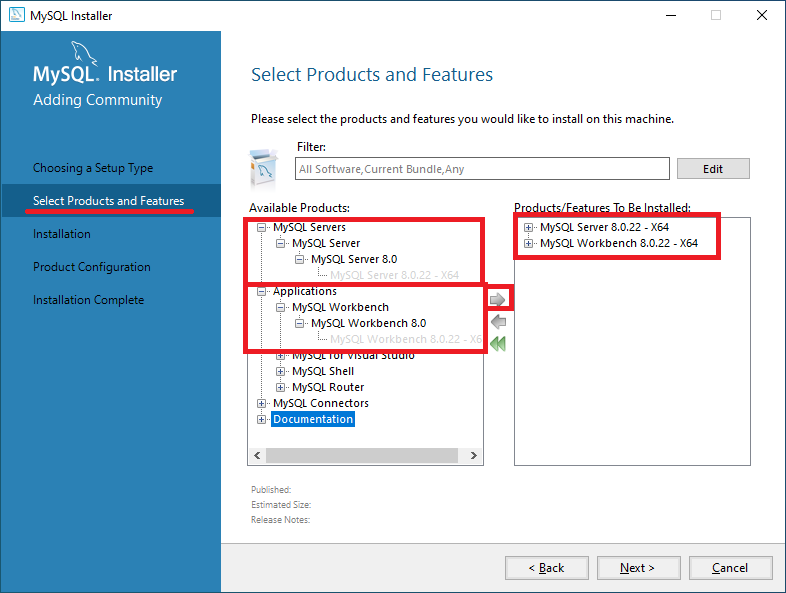
### Configuration du serveur MySQL

A ce moment nous configurons notre serveur.

Dans « Choosing a Setup Type », choisir « Custom ». Puis appuyer sur « Next ».



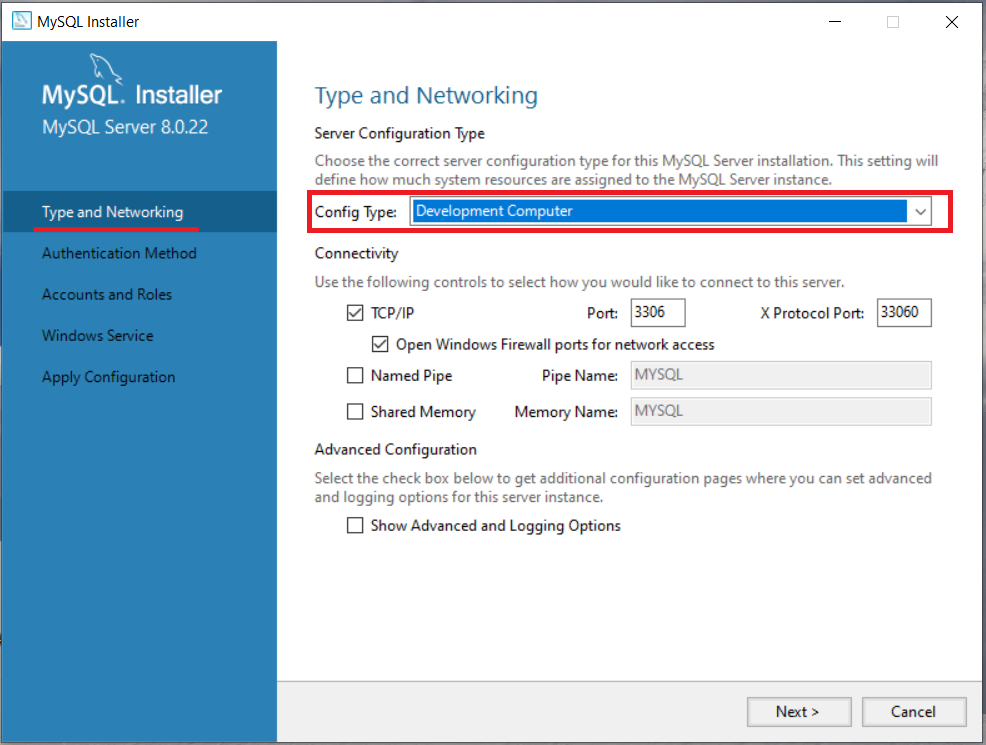
Ensuite, dans « Select Products and Features », sélectionner MySQL SAerver 8.0 et cliquer sur la flèche en direction de droite. Faite pareil en sélectionnant « MySQL Workbench 8.0 », puis clique sur « Next ».



Sous « Installation » cliquer sur « Exécute ». Appuyer ensuite sur « Next ».

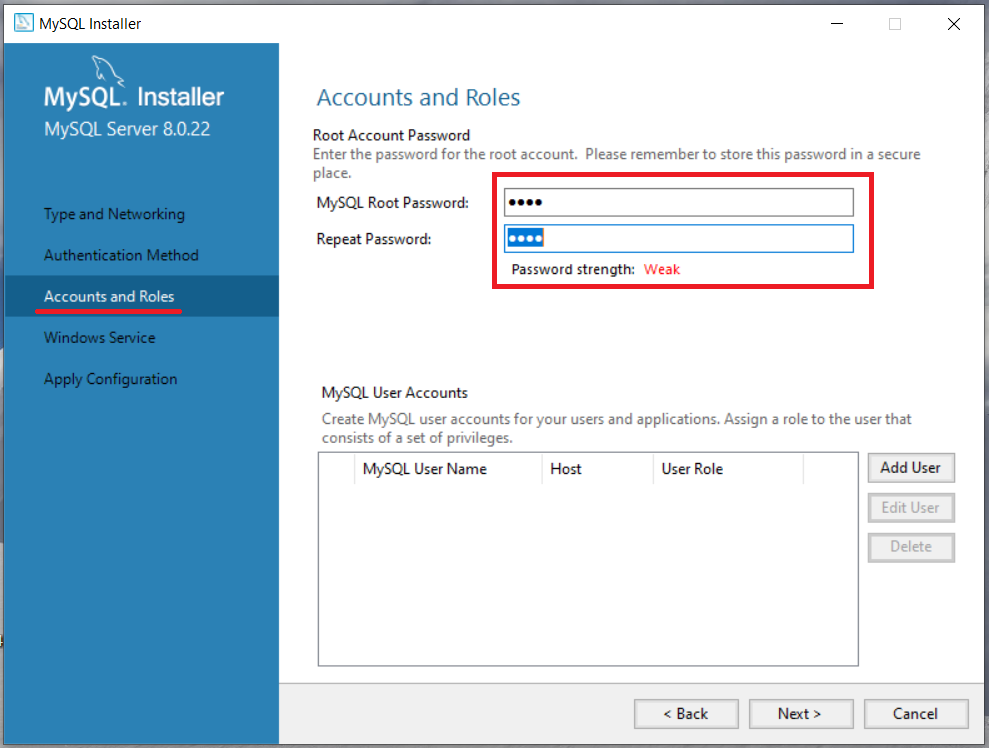
Sous « Product Configuration », cliquer encore sur « Next ».

Dans « Type and Networking », mettre « Config Type » en « Development Computer » puis tout laisser par défaut. Appuyer sur « Next ».

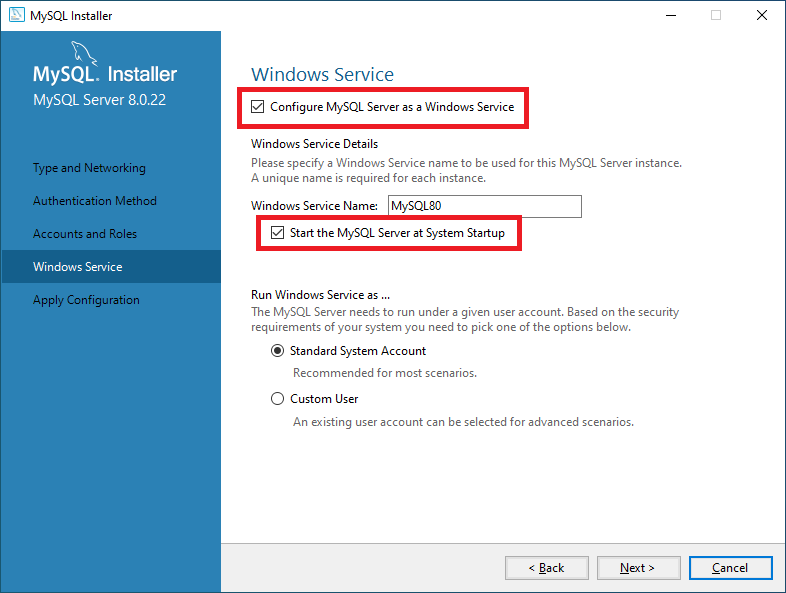


Dans l’onglet « Authentification Method », garder les paramètres par défaut et appuyer sur « Next ».

Dans l’onglet « Accounts and Roles », entrer deux fois le mot de passe du compte administrateur souhaité et appuyer sur « Next ».



Dans « Windows Service » cocher « Start the MySQL Server at System Startup ». De plus, il faut aussi vérifier que « Configure MySQL Server as a Windows Service » soit bien sélectionné. Puis appuyer encore sur « Next ».



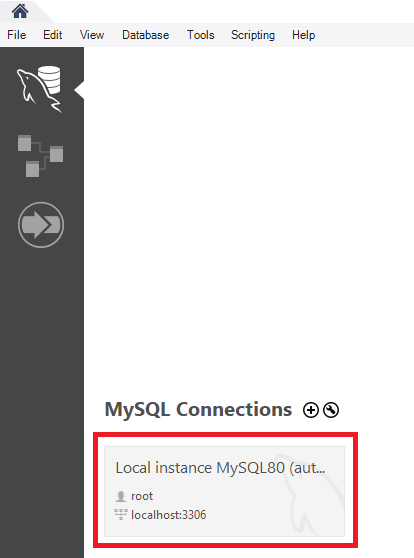
Finalement dans « Apply Configuration », cliquer sur « Execute ».

Aller jusqu’à la fin de la configuration et de l’installation en laissant tout par défaut puis terminer en cliquant sur « Finish ».

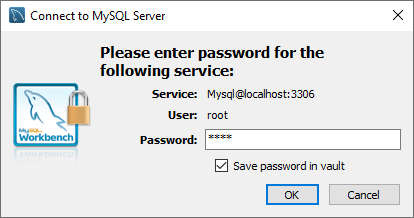
Désormais, vous avez MySQL Workbench lancé. Si ce n’est pas le cas lancer le programme en appuyant sur la touche « Windows », puis en tapant MySQL Workbench dans la barre de recherche et en appuyant finalement sur « Enter ».

### Installation de la base de données

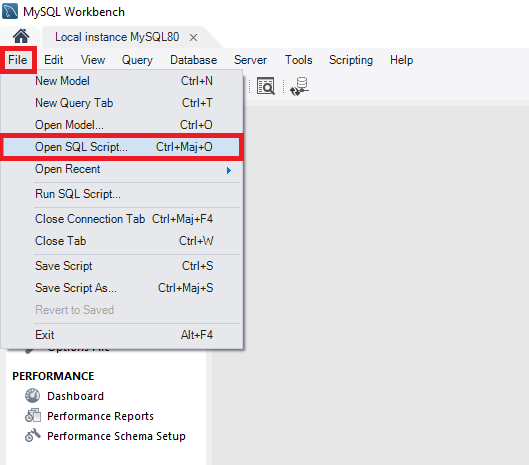
Dans MySQL Workbench, double cliquer sur « Local instance … ».



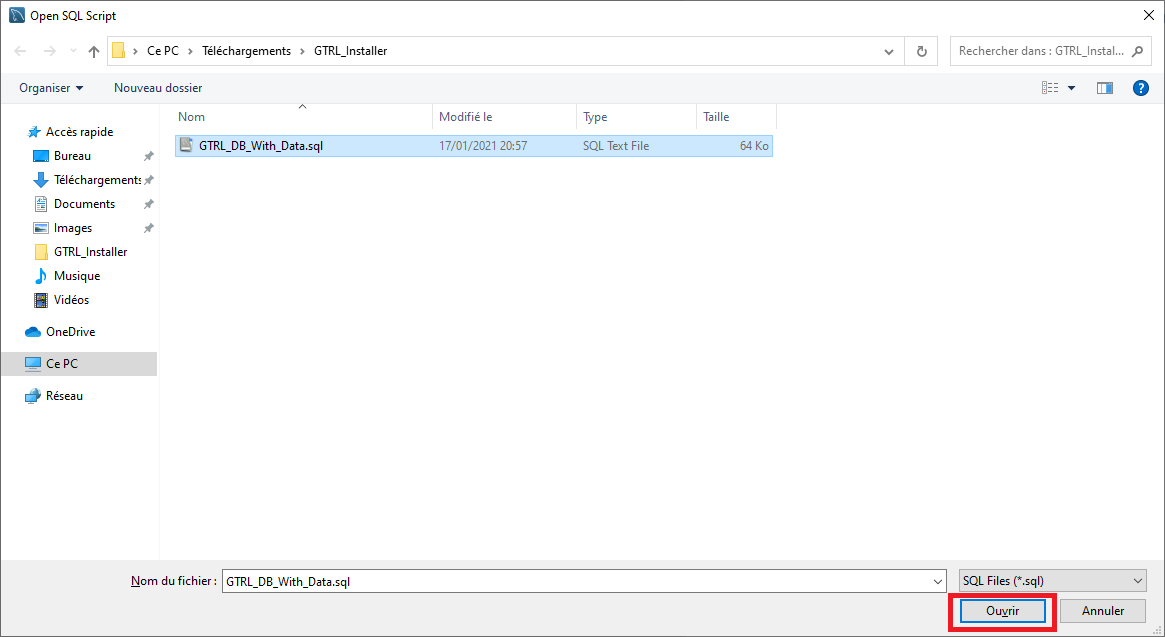
Entrer ensuite votre mot de passe configurer plus tôt.



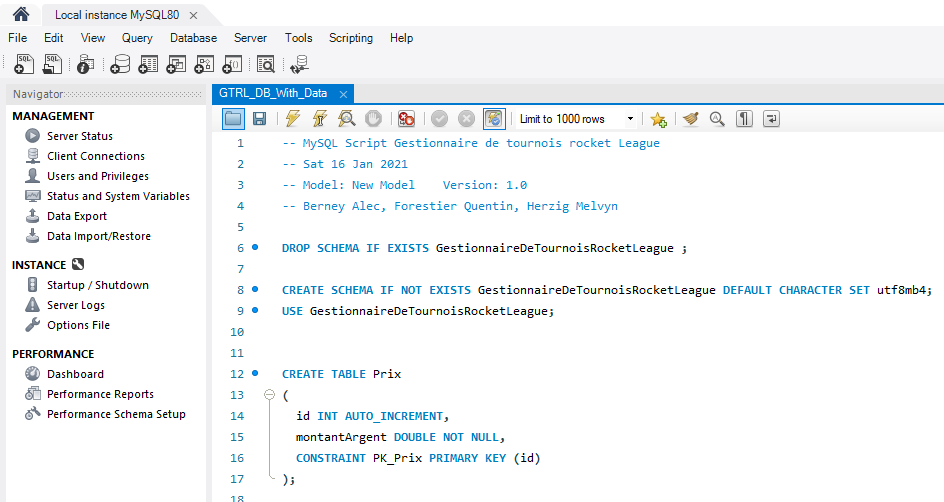
Clique sur « File », puis « Open SQL Script ».



Sélectionner ensuite « GTRL\_DB\_With\_Data.sql » dans le dossier extrait du zip où était situé l’installeur, vous trouverez. Cliquer ensuite sur « Ouvrir », cela installera la base de données avec une jeu de données préexistantes. Sinon pour une installation vierge, utiliser le script « GTRL\_DB.sql ».

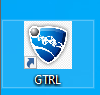
****

Vous voilà avec le script ouvert, vous devez avoir affiché quelque chose comme ça :

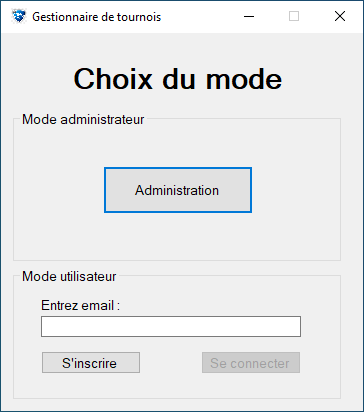


Cliquer ensuite sur le petite éclair .

Vous pouvez mnt fermer MySQL Workbench. Vous pouvez tester le programme en double cliquant sur le raccourci sur votre bureau :



Voici le programme lancé :



Pour connaître comment utiliser le programme, suivre le manuel utilisateur de la documentation.

## Fonctions stockées

|  |
| --- |
| BOOLEAN aUneInscriptionEnCours(pAcronymeParam VARCHAR(3)) |
| pAcronymeParam : Acronyme de l’équipe à vérifier. |
| Retourne : TRUE si l’équipe en paramètre est inscrite à un tournoi « En attente » sinon FALSE. |

Vérifie si l’équipe passé en paramètre est déjà inscrite à un tournoi à venir.

|  |
| --- |
| INT calculerNbTours(pNbEquipeMax INT) |
| pNbEquipeMax : Le nombre maximal d’équipe du tournoi à vérifier. |
| Retourne : Le nombre de tour pour faire joueur les équipes, 0 si paramètre invalide. |

Pour le nombre maximal d’équipe passé en paramètre vérifie s’il correspond à une puissance de 2 entre 2 et 256 et retourne le nombre de tour que cela représente sinon 0.

|  |
| --- |
| INT compterVictoireDansSerie (pAcronymeEquipe VARCHAR(3), pIdSerie INT, pNoTour INT, pIdTournoi INT,   pSignalerMatchErreur BOOLEAN) |
| pAcronymeEquipe : Équipe à compter les victoires. |
| pIdSerie : Numéro de la série. |
| pNoTour : Numéro du tour. |
| pIdTournoi : Numéro du tournoi. |
| pSignalerMatchErreur : Indique si l’appel de vainqueurMatch peut lever des signaux |
| Retourne : Le nombre de victoire de l’équipe pour la série donnée. |

Pour une équipe et une série donnée compte son nombre de victoire. Si l’équipe ne faisait pas partie de la série ou qu’elle n’a tout simplement pas gagné, retourne 0.

|  |
| --- |
| VARCHAR(3) equipeDuJoueurLorsDu(pIdJoueur INT, pDate DATETIME) |
| pIdJoueur : Numéro du joueur à rechercher. |
| pDate : Date à laquelle on veut connaître son équipe. |
| Retourne : L’acronyme de son équipe s’il en avait une, sinon NULL |

Pour un joueur donné, recherche à quelle équipe il appartenait à une date donnée.

|  |
| --- |
| BOOLEAN estComplete(pAcronymeEquipe VARCHAR(3)) |
| pAcronymeEquipe : Équipe à vérifier. |
| Retourne : TRUE si l’équipe est complète sinon FALSE. |

Vérifie si l’équipe est complète.

|  |
| --- |
| BOOLEAN estEnAttente(pIdTournoi INT) |
| pAcronymeEquipe : Tournoi à vérifier. |
| Retourne : TRUE si la date de début n’est pas encore atteinte. |

Vérifie si le tournoi est en attente.

|  |
| --- |
| BOOLEAN seedingCorrect (pAcronymeEquipe1 VARCHAR(3), pAcronymeEquipe2 VARCHAR(3), pIdSerie INT, pNoTour INT, pIdTournoi INT) |
| pAcronymeEquipe1 : Acronyme de l’équipe qui devrait être en acronymeEquipe1 de la série. |
| pAcronymeEquipe2 : Acronyme de l’équipe qui devrait être en acronymeEquipe2 de la série. |
| pIdSerie : Numéro de la série. |
| pNoTour : Numéro du tour. |
| pIdTournoi : Numéro du tournoi. |
| Retourne : TRUE si les équipes seraient à leur place sinon FALSE. |

Vérifie si les équipes en paramètre peuvent occuper acronymeEquipe1 et acronymeEquipe2 de la série en référence. La vérification est faite selon le chapitre « Seeding » de ce document.

|  |
| --- |
| VARCHAR(3) vainqueurSerie (pIdSerie INT, pNoTour INT, pIdTournoi INT, pSignalerMatchErreur BOOLEAN) |
| pIdSerie : Numéro de la série. |
| pNoTour : Numéro du tour. |
| pIdTournoi : Numéro du tournoi. |
| pSignalerMatchErreur : Indique si l’appel de vainqueurMatch peut lever des signaux |
| Retourne : L’acronyme de l’équipe qui a gagné la série, en cas d’impossibilité retourne NULL. |

Recherche le vainqueur de la série donnée. Si le vainqueur ne peut être déterminé, la fonction retourne NULL.

|  |
| --- |
| VARCHAR(3) vainqueurMatch (pIdMatch INT, pIdSerie INT, pNoTour INT, pIdTournoi INT, pSignalerMatchErreur BOOLEAN) |
| pIdMatch : Numéro du match. |
| pIdSerie : Numéro de la série. |
| pNoTour : Numéro du tour. |
| pIdTournoi : Numéro du tournoi. |
| pSignalerMatchErreur : Indique si la fonction peut lever des signaux |
| Retourne : L’acronyme de l’équipe qui a gagné le match. |
| Signal : - Equipes égalités.  - Certains des 6 joueurs n’ont pas d’enregistrement Match\_Joueur associé. |

Recherche le vainqueur du match donné.

|  |
| --- |
| BOOLEAN estComplet(pIdTournoi INT) |
| pIdTournoi : Numéro du tournoi à vérifier si complet. |
| Retourne : Vrai si le nombre maximal d’équipes inscrite est atteint. |

Renvoie vrai si toutes les places du tournoi sont occupées.

|  |
| --- |
| BOOLEAN seedingEffectue(pIdTournoi INT) |
| pIdTournoi : Numéro du tournoi à vérifier si complet. |
| Retourne : Vrai les équipes inscrites ont été placées dans une série du premier tour du tournoi. |

Vérifie si les séries du premier tour du tournoi sont composées.(celui avec l’ensemble des équipes)

## Procédures stockées

Tous les signaux listés ont l’état SQLSTATE ‘45000’ qui correspond à une exception utilisateur.

|  |
| --- |
| verifierDateFuture(pDate DATETIME) |
| pDate: Date à vérifier. |
| Signal : - La date est strictement plus grande que le temps courant. |

Vérifie si la date est dans le futur. Dans ce cas, un signal est levé.

|  |
| --- |
| verifierDatePassee(pDate DATETIME) |
| pDate: Date à vérifier. |
| Signal : - La date est plus petite ou égale au temps courant. |

Vérifie si la date est passé ou présente. Dans ce cas, un signal est levé.

|  |
| --- |
| verifierDatePlusPetite(pDateHeureDebut DATETIME, pDateHeureFin DATETIME) |
| pDateHeureDebut: Date heure de début. |
| pDateHeureFin : Date heure de fin. |
| pNoTour : Numéro du tour. |
| Signal : - La date arrivée est plus grande que la date départ |

Vérifie si la date d’arrivée est strictement plus grande que de départ. Dans ce cas, un signal est levé.

|  |
| --- |
| verifierDejaDansUneEquipe(pIdJoueur INT) |
| pIdJoueur : Joueur à rechercher. |
| Signal : - Si l’équipe est membre actuel d’une équipe |

Vérifie si le joueur fait partie d’une équipe et lève un signal.

|  |
| --- |
| verifierEquipeComplete(pAcronymeEquipe VARCHAR(3)) |
| pAcronymeEquipe : Équipe à vérifier. |
| Signal : - Si l’équipe est complète. |

Vérifie si l’équipe contient trois joueurs, si c’est le cas, un signal est levé.

|  |
| --- |
| verifierInscription(pAcronyme VARCHAR(3), pIdTournoi INT) |
| pAcronyme : Équipe à vérifier. |
| pIdTournoi : Numéro du tournoi correspondant. |
| Signal : - Si l’équipe n’est pas inscrite au tournoi. |

Vérifie si l’équipe ne fait pas partie des inscrites pour le tournoi donné et lève un signal.

|  |
| --- |
| verifierJoueurEstDansEquipeSerie(pIdJoueur INT, pIdSerie INT, pNoTour INT, pIdTournoi INT) |
| pIdJoueur : Joueur à vérifier la légitimité |
| pIdSerie : Numéro de la série. |
| pNoTour : Numéro du tour. |
| pIdTournoi : Numéro du tournoi. |
| Signal : - Si le joueur ne fait partie d’aucune équipe au moment du tournoi.  - Si le joueur appartient à une équipe qui ne dispute pas la série. |

Vérifie si un joueur peut légitiment avec des résultats dans la série associée.

|  |
| --- |
| verifierLongueurMaxSerie(pLongueurSerie INT) |
| pLongueurSerie : Nombre maximal de match de la série. |
| Signal : - Le nombre n’est pas 1, 3, 5, 7. |

Vérifie si la longueur maximale de série est un nombre autorisé.

|  |
| --- |
| verifierMemeEquipe(pAcronymeEquipe1 VARCHAR(3), pAcronymeEquipe2 VARCHAR(3)) |
| pAcronymeEquipe1 : Nom d’une équipe. |
| pAcronymeEquipe2 : Nom d’une équipe. |
| Signal : - Les deux équipes ont le même nom et sont donc les mêmes. |

Vérifie si les deux équipes sont les même et lève un signal.

|  |
| --- |
| verifierSeedingIncorrect (pAcronymeEquipe1 VARCHAR(3), pAcronymeEquipe2 VARCHAR(3), pIdSerie INT, pNoTour INT, pIdTournoi INT) |
| pAcronymeEquipe1 : Acronyme de l’équipe qui devrait être acronymeEquipe1 dans la série. |
| pAcronymeEquipe2 : Acronyme de l’équipe qui devrait être acronymeEquipe2 dans la série. |
| pIdSerie : Numéro de la série. |
| pNoTour : Numéro du tour. |
| pIdTournoi : Numéro du tournoi. |
| Signal : - Le seeding ne peut pas être respecté avec une des deux équipes. |

Vérifie si le seeding est respecté si les deux équipes sont placées dans la série correspondante sinon lève un signal.

|  |
| --- |
| verifierTournoiEnAttente(pIdTournoi INT) |
| pIdTournoi : Tournoi à vérifier. |
| Signal : - Le tournoi n’est pas en attente. |

Vérifie si le tournoi est en attente, dans le cas contraire, lève un signal.

|  |
| --- |
| verifierAcronyme(pAcronymeEquipe VARCHAR(3)) |
| pAcronymeEquipe: Acronyme d’une équipe. |
| Signal : - L’acronyme ne fait pas au moins un caractère. |

Vérifie si la longueur de l’acronyme est valide, qu’il fasse au moins 1 cractère.

|  |
| --- |
| demarrerTournoi(pIdTournoi INT) |
| pIdTournoi: Tournoi à démarrer. |
| Signal : - Le tournoi est encore en attente.  - Il n’y a pas assez d’équipes |

Démarre un tournoi, dispose les équipes inscrites dans le premier tour du tournoi.

|  |
| --- |
| tenterPromotion (pIdMatch INT, pIdSerie INT, pNoTour INT, pIdTournoi INT) |
| pIdMatch: id du match |
| pIdSerie : id de la série du match. |
| pNoTour : id du Tour de la série |
| pIdTournoi : Tournoi d’appartenance du tour |

Pour le match donné, dès que les 6 joueurs ont été enregistrés, contrôle si la série est terminée. Si c’est le cas, le vainqueur est placé dans la série correspondante suivante.

|  |
| --- |
| verifierEmail(pEmail VARCHAR(250)) |
| pEmail: Email à vérifier. |
| Signal : - Le courriel n’a pas le bon format |

Vérifie si le courriel valide.   
Expression régulière : '^[a-zA-Z0-9][a-zA-Z0-9.\_-][\*@[a-zA-Z0-9][a-zA-Z0-9.\_-]\*\.[a-zA-Z]{2,4}$](mailto:*@[a-zA-Z0-9][a-zA-Z0-9._-]*\.%5ba-zA-Z%5d%7b2,4%7d$)'

|  |
| --- |
| verifierInscriptionEquipe(pAcronymeEquipe VARCHAR(3), pIdTournoi INT) |
| pAcronymeEquipe : Équipe à inscrire. |
| pIdTournoi : Tournoi d’inscription. |
| Signal : - Le tournoi n’est plus en attente.  - Le tournoi est complet.  - L’équipe est déjà inscrite à un tournoi.  - L’équipe n’est pas complète. |

Effectue les vérifications des conditions d’inscription d’une équipe à un tournoi.

|  |
| --- |
| verifierPrixNegatif(pPrix DOUBLE) |
| pPrix: Prix à vérifier. |
| Signal : - Le prix est négatif. |

Vérifie si le prix passé en paramètre est négatif.

|  |
| --- |
| accepterJoueur(pAcronymeEquipe VARCHAR(3), pIdJoueur INT) |
| pAcronymeEquipe: Acronyme de l’équipeà rejoinder. |
| pIdJoueur: Identifiant du joueur à accepter. |
| Signal : - L’équipe est pleine |

Effectue la transaction pour rejoindre une équipe : acceptation de la demande et suppression des autres.

## Événements

Deux événements ont été mis en place afin de détecter et supprimer les tournois qui n’aurait pas assez d’équipes au moment où ils devraient commencer.

|  |
| --- |
| annuler\_tournoi |
| Quand Toutes les heures. |
| Qui : Tous les tournois dont la date de début est dépassée et qui n’ont pas assez d’équipes. |
| Effet : Initialise la date heure de fin à leur date heure de début. |

|  |
| --- |
| supprimer\_tournoi\_annule |
| Quand Tous les 7 jours. |
| Qui : Tous les tournois qui ont la date heure de début égale à date heure de fin |
| Effet : Suppression des tournois. |

La première activation à lieu 15 minutes après sa création. Ceci a pour but de ne pas supprimer instantanément les tournois de test annulés.

## Application des contraintes d’intégrité

Les différentes contraintes d’intégrité ont été résolues avec des triggers. Par simplification nous dirons **[insertion]**, **[mise à jour]**, **[suppression]** pour parler de triggers qui interviennent à l’insertion, mise à jour et suppression de champs aux tables concernées.

**Tournoi**

Le nombre d’équipe d’un tournoi = 2^x ou x <- [1, 2 .. 8]

[Insertion]: Vérification que le log₂ du nombre maximal d’équipe soit un entier entre 1 et 8.  
 [Mise à jour] : Le nombre maximal d’équipe et inchangeable mais le tournoi peut être  
 supprimé tant qu’il n’a pas commencé.

Le champ dateHeureDebut est strictement plus petite que dateHeureFin

[Insertion] La date de fin doit être nulle.  
[Mise à jour] : Appel de la procédure verifierDatePlusPetite.

Le champ dateHeureDebut est strictement plus grande que la date courante

[Insertion, Mise à jour] : Appel de la procédure verifierDatePassee.

**Tournoi\_Equipe**

La date d’inscription d’une équipe est plus vieille que la dateDebut du tournoi

[Insertion, Mise à jour] : Appel de la procédure verifierTournoiEnAttente.

Une équipe ne peut pas s’inscrire à un tournoi si elle n’a pas trois joueurs

[Insertion, Mise à jour] : Appel de la procédure verifierEquipeComplete.

Ne peut pas s’inscrire à un nouveau tournoi tant qu’elle est inscrite à un tournoi en « attente » ou « en cours »

[Insertion, Mise à jour] : Appel de la procédure aUneInscriptionEnCours.

Peut seulement s’inscrire à un tournoi « en attente »

[Insertion, Mise à jour] : Appel de la procédure verifierTournoiEnAttente.

**Tour**

Le champ longueurMaxSerie = [1,3,5,7].

[Insertion, Mise à jour] : Appel de la procédure verifierLongueurMaxSerie.

**Serie**

Référence deux équipes inscrites au tournoi associé et non éliminées.

[Insertion, Mise à jour] : Appel de la procédure verifierInscription pour les deux équipes. Appel de la procédure verifierMemeEquipe.  
 Appel de la procédure verifierSeedingCorrect.

Le nombre de buts total des deux équipes ne peuvent pas être égaux.

[Insertion, Mise à jour] : Lors de l’appel à la procédure vérifierSeedingCorrect, si un des   
 deux matchs précédents la série sont nuls, un signal est levé par la   
 fonction vainqueurMatch. La final est vérifiée lorsqu’on termine  
 le tournoi y mettant une date de fin.

Une équipe ne peut pas gagner si elle joue contre aucune autre.

[Insertion, Mise à jour] : Appel de verifierSeedingIncorrect. Pour que l’équipe soit ajoutée,  
 elle doit avoir gagné la série précédente, ce qui est vérifié avec  
 vainqueurSerie qui n’accepte pas que la série précédente n’ait pas  
 deux équipes. Après le 6ème enregistrement Match\_Joueur,  
 si c’est la finale, on vérifie s’il y a un vainqueur dans la série avec  
 vainqueurSerie, si oui le tournoi est terminé.

**Match\_Joueur**

Référence un joueur qui joue pour une des deux équipes de la série

[Insertion, Mise à jour] : Appel de la procédure vérifierJoueurEstDansEquipeSerie.

Les buts et les arrêts sont positifs ou nuls

Le type de données des buts et des arrêts est entier non signé. La valeur par défaut est 0.

Un joueur doit avoir des buts/arrêts que dans un match auquel il a participé

[Insertion, Mise à jouer] : Appel de la procédure verifierJoueurEstDansEquipeSerie.

**Equipe**

Le responsable est forcément un joueur

[Insertion] : Après insertion crée automatiquement un enregistrement dans Equipe\_Joueur.  
 [Mise à jour] : Si le responsable tente d’être changé, l’opération est annulée.

**Joueur**

La date de naissance doit être plus petite que la date courante

[Insertion, Mise à jour[ : Appel à la procédure verifierDateFuture.

L’email doit être unique

Ajout d’une contrainte d’unicité sur le champ.

**Equipe\_Joueur**

Le champ dateHeureArrivee est forcément plus petite que dateHeureDepart

[Insertion, Mise à jour] : Appel de la procédure vérifierDateArriveeDepart.

Un enregistrement ne peut être créé pour un joueur tant qu’il est déjà dans une équipe

[Insertion] : Appel de la procédure estDejaDansUneEquipe.

Ne peut pas quitter son équipe tant qu’elle participe à un tournoi

[Mise à jour] : Appel de la fonction aUneInscriptionEnCours, lève un signal si nécessaire.

Un joueur ne joue que dans une équipe à la fois

[Insertion] : Appel de la procédure verifierDejaDansUneEquipe.

**Objet**

Le nom doit être unique

Ajout d’une contrainte d’unicité sur le champ.

**Prix**

Le prix n’est pas négatif

[Insertion, Mise à jour] Appel à la procédure verifierPrixNegatif.

## Comportements mis en place

**Tournoi**

La suppression d’un tournoi est possible que s’il n’a pas débuté ou qu’il est annulé.

Le nombre d’équipes du tournoi ne peut pas être modifié afin de ne pas avoir à gérer dynamiquement l’arbre de tournoi dans la base de données.

La date de début du tournoi peut être modifiée tant que le tournoi est en attente.

La date de fin d’un tournoi ne peut être initialisée que si la finale a été disputée. Il n’est pas possible d’initialiser la date de fin à NULL une fois qu’elle a été initialisée à une date.

**Tournoi\_Equipe**

L’ajout d’une nouvelle entrée n’est possible que si le nombre maximal d’équipes du tournoi n’est pas atteint, qu’il est toujours en cours, que l’équipe soit complète et non inscrite à un autre tournoi. Un enregistrement est supprimable que si le tournoi est en attente. Les dates d’inscriptions sont automatiquement initialisées au temps courant.

**Tour**

Un tour ne peut pas être modifié si le tournoi a déjà débuté.

Les tours sont créés automatiquement quand un tournoi est inséré.

Un tour ne peut pas être supprimé tant que le tournoi ne l’est pas.

Les tours doivent être générés en commençant depuis la finale, faut de quoi le no sera invalide.

**Série**

A l’insertion et à la modification, nous vérifions que les deux équipes concernées ne sont pas identiques.  
Les équipes qui peuvent joueur dans la série doivent respecter le système de seeding   
(Cf. : chapitre « Seeding ») mis en place. De cette manière, nous assurons que les équipes sont inscrites au tournoi et non éliminées.

Les séries sont créées automatiquement quand un tournoi est inséré.

Les équipes sont promues automatiquement entre les séries. A chaque fin de match une vérification automatique est effectuée.

Une série ne peut pas être supprimée tant que le tour associé ne l’est pas.

Equipe

Le responsable est automatiquement inséré comme joueur et ses autres demandes sont effacées.

**Equipe\_Joueur**

À l’insertion seul les demandes sont acceptées.

À la mise à jour, si c’est une acceptation, la date et initialisée au temps courant.

La modification est très restreinte, seuls les dates peuvent être modifiées et sous conditions.

Un joueur a une demande d’inscription dans une équipe si la date   
d’arrivée est ‘0001-01-01 00 :00 :00’

Un joueur est officiellement dans une équipe lors la date d’arrivée   
n’est plus ‘0001-01-01 00 :00 :00’ est que la date de départ est nulle.

Un joueur a quitté une équipe lorsque la date de départ n’est plus nulle.

En utilisant la procédure accepterJoueur, si un joueur est accepté dans une équipe toutes ses autres demandes sont supprimées.

**Joueur**

Une expression régulière a été mise en place pour vérifier le format de l’adresse mail.

**Match**

Un match ne peut être créer que s’il ne dépasse pas la limite de match possible et que le gagnant de la série ne peut être déterminé.

Un match ne peut être supprimé si la série associée est toujours existante.

**Match\_joueur**

Si le match contient 6 enregistrements, le match est considéré comme terminé et on vérifie l’état de la série. Si c’est la finale, la date de fin de tournoi est initialisée, sinon on tente de promouvoir une des deux équipes si elle a remporté la série associée.

Les enregistrements ne peuvent pas être modifiés si la série suivante a déjà commencé. Cela risquerait d’entrainer des incohérences.

## Seeding

Le « seeding » est système qui définit l’emplacement de chaque équipe dans l’arbre de tournoi en fonction de ses résultats.

**Premier tour :**

Les équipes sont groupées par ordre des dates d’arrivée. Ainsi dans l’exemple, les deux équipes inscrites depuis le plus longtemps sont E1 et E2. E1 étant la plus vieille, elle sera l’équipe 1 et E2 l’équipe2. De même pour E3 et E4, E3 sera l’équipe 1 et E4 l’équipe 2.

**Tours suivants :**

Pour les tours qui suivent, l’opérations est la suivante :

Soit T le numéro de tour.

Soit nS le numéro de la série au tour T.

L’équipe 1 est le vainqueur de la série de numéro nS \* 2 – 1 du tour T + 1.  
L’équipe 2 est le vainqueur de la série de numéro nS \* 2 du tour T + 1.



Figure Progression dans un arbre de tournoi.