

Mamadou Yaya DIALLO  
Natchley CANON  
Pierrick HERVE  
Serigne DIAW

## Projet BD/WEB : Mastermind

### BD

#### Schéma entité association

(voir annexe)

#### Schéma relationnel

(voir annexe)

#### Description de contraintes colonne-table

Nous avons mis une contrainte Not Null pour la plupart des attributs afin d'avoir des données "propres" dès les insertions. Pour les attributs suivants : expérience du joueur, catégorie du joueur, résultat de la partie et score de la partie nous avons défini une valeur par défaut.

Nous avons utilisé deux contraintes Check, une pour obliger le joueur à choisir un mot de passe d'au moins 8 caractères, et une autre pour restreindre les valeurs de resultatP à 0 ou 1 selon si la partie est perdue ou gagnée.

#### Liste de triggers et explication de leur objectifs

Vous trouverez ci-dessous la liste des triggers existants et leur objectif.

Trigger 1 : t_a_i_Joueur
On débloque le niveau 1 pour tous les nouveaux joueurs.

<b>Trigger 2 : t_b_i_Partie</b>
---------------------------------

<p>On vérifie que le nombre de billes jouables est supérieur ou égal au nombre d'emplacements à remplir. En effet, si ce n'est pas le cas et étant donné qu'on ne peut jouer la même bille qu'une fois par ligne, le joueur ne pourra alors soumettre aucune proposition.</p>
---

<b>Trigger 3 : t_b_i_Partie</b>
---------------------------------

<p>On empêche le joueur de créer des parties pendant quatre heures si il a perdu 5 parties en une heure.</p>
--

<b>Trigger 5 : t_b_i_Partie</b>
---------------------------------

<p>On vérifie que le joueur ai suffisamment d'expérience pour pouvoir jouer au niveau qu'il a sélectionné, si c'est le cas et que la ligne . Normalement l'accès aux niveaux est géré dans la table "Débloquer" mais dans le cas où un joueur est créée avec une expérience de départ non nulle, la table "Débloqué" contient uniquement l'accès au niveau 1 (créée par le Trigger 1).</p>
--

<b>Trigger 6 : t_b_i_multiJ_Partie</b>
--

<p>Dans le cas des parties en multi-joueurs on vérifie que le joueur ayant le moins d'expérience puisse accéder au niveau sélectionné. Si ce n'est pas le cas on annule la création de la partie.</p>
---

<b>Trigger 7 : t_b_u_expJ</b>
-------------------------------

<p>A chaque fois que l'expérience d'un joueur est mise à jour, on vérifie qu'elle n'ai pas diminuée, car un joueur ne peut pas perdre d'expérience.</p>
---

<b>Trigger 8 : t_b_u_catJ</b>
-------------------------------

<p>A chaque fois que la catégorie d'un joueur est mise à jour, on vérifie qu'il n'ai pas été déclassé.</p>
--

<b>Trigger 9 : t_a_u_partie</b>
---------------------------------

<p>Si le joueur gagne une partie on lui débloque le niveau suivant en insérant une ligne dans la table "Débloquer".</p>
---

<b>Trigger 10 : t_b_i_proposition_joueur</b>
Vérifie si la ligne est déjà complète et si la bille proposée par le joueur a déjà été jouée.

Nous avons fait le choix de ne pas fusionner les triggers portant sur les mêmes tables (partie, joueur, ...) et au même moment (before ou after) car ils auraient été beaucoup moins lisibles. De plus il n'y a pas de conflit entre les différents traitements réalisés par ces triggers.

(code voir annexe)

## Liste de procédures et explication de leur objectifs

Vous trouverez ci-dessous la liste des procédures existantes et leur objectif.

<b>Procédure 1 : P_creation_combinaison</b>
On crée une combinaison avec aucune bille en double à partir de la Collection de bille correspondante au niveau joué.

<b>Procédure 2 : Rejouer</b>
Rejoue une partie en affichant les lignes dans l'ordre dans lesquelles elle ont été jouées, une toutes les secondes.

<b>Procédure 3 : Insertion_joueur</b>
Gestion de l'insertion des joueurs lors de la création d'un compte

<b>Procédure 4 : Insertion_partie</b>
Gestion de l'insertion d'une partie. Par la suite, il faut insérer une ligne dans la table Jouer.

<b>Procédure 5 : Insertion_partie_multi</b>
Gestion de l'insertion d'une partie dans le cadre d'une partie multijoueur.

<b>Procédure 6 : Insertion_ligne</b>
Gestion de l'insertion d'une ligne pour une partie donnée.

<b>Procédure 7 : CalculScore</b>
Gestion du calcul du score d'un joueur à la fin d'une partie.

## Description des droits par utilisateur de la BD

Deux comptes distincts existent pour la BD :

- compte ADMIN utilisé pour administrer la BD
- compte USERS utilisé pour jouer au Mastermind

Le compte ADMIN a tous les droits, tandis que le compte USERS à des droits restreints sur les tables comme indiqué dans le tableau ci-dessous.

On autorise les updates sur la table partie pour pouvoir mettre à jour le score du joueur une fois la partie terminée.

Table	Actions interdites (sécurité des données)
Composer	restriction_1
Jouer	restriction_1
Proposition_Joueur	restriction_2 (avec update)
Debloquer	restriction_1
Combinaison	restriction_1
Ligne	restriction_2 (avec update)
Partie	restriction_2 (avec update)
Joueur	restriction_1
Niveau	restriction_1
CategorieJoueur	restriction_1
Bille	restriction_1
Collection	restriction_1

Légende :

- restriction\_1: update, delete, index, references
- restriction\_2 (avec update) : delete, index, references

## Problèmes rencontrés côté BD

- implémentation des highscores en utilisant des vues
- typage des données exploitable en WEB
- correction des triggers

## WEB

### Architecture globale

(voir annexe)

### Architecture “fonctionnelle” d'une partie monojeueur

(voir annexe)

### Architecture “fonctionnelle” d'une partie multijoueur

(voir annexe)

## Problèmes rencontrés côté WEB

- interaction entre les différents langages JavaScript, PHP => affichage de JavaScript grâce à PHP (commande “echo”)
- agencement des éléments au sein des pages => rapidement résolu à l'aide des “flex”
- transfert de données entre les pages => GET et POST
- rigueur dans le choix des attributs très utilisés (ex: id, class) => commentaire, tabulation

# Gestion du projet

## Répartition du travail et temps passé

Pour organiser le travail nous avons utilisé l'outil "Trello" qui consiste en un tableau KANBAN interactif dans lequel nous avons identifié grâce à des couleurs les tâches relevant de la BD et celle relevant du WEB.

Nous avons cherché au maximum à répartir le travail pour que tous les membres du groupe travaille à la fois sur le côté WEB et le côté BD. C'est le service GitHub qui nous a permis de mener à bien cet objectif.

Au début du projet, chaque membre du groupe a travaillé sur la partie du projet correspondant à ses préférences puis nous avons tourné aux différents postes : administration BD, développement WEB, génération des données.

En début de projet nous avons fait beaucoup de réunions pour être sûr de partager la même vision du projet, mais aussi de concevoir la BD.

Nous avons chacun consacré en moyenne 4h00 par semaine en plus des 2h00 de TP encadrées au projet.

## Bilan final

En conclusion on peut dire que le projet nous a beaucoup apporté, tant en terme de compétences acquises qu'en terme de savoir-faire et de savoir-être. Vous trouverez ci-après la liste de ses points principaux. Les principales compétences que nous avons acquises sont les suivantes, prise en main des technologies GitHub, Virtualbox et des langages PHP, Ajax, JavaScript. Concernant le savoir-faire nous avons pris conscience de l'importance de l'entraide au sein du groupe notamment entre les membres du groupe ayant des profils orientés WEB et les autres orientés BD. De plus, nous avons su bien communiquer et cela a été indispensable au bon avancement du projet, nous avons donc pu constater que la communication est essentielle. Enfin, pour ce qui est du savoir-être, nous avons développé notre sens de l'écoute.

Une chose particulièrement importante est le fait que nous ayant tourné aux différents postes (administration BD, développement WEB, génération des données), bien que cela nous ait ralenti, cela a permis à tous les membres du groupes de gagner en polyvalence ce qui est un plus non négligable