Rapport Base de données

Rapport concernant base de données



**Lucas LEVEQUE, Berachem MARKRIA & Joshua LEMOINE**

17/12/2022

BUT2 Informatique

# Introduction

Ce rapport décrit la base de données que nous utiliserons pour le projet Boggle (Zoo-ggle). L’objectif de cette base de données est pour nous de pouvoir stocker les informations concernant les joueurs ( les interactions entre eux via un chat privé), les parties et leurs conditions de jeu (grille, mode, dictionnaire, public ou non …) lors de leur déroulement (créateur, mot proposés par les joueurs, score, chat de la partie… ).

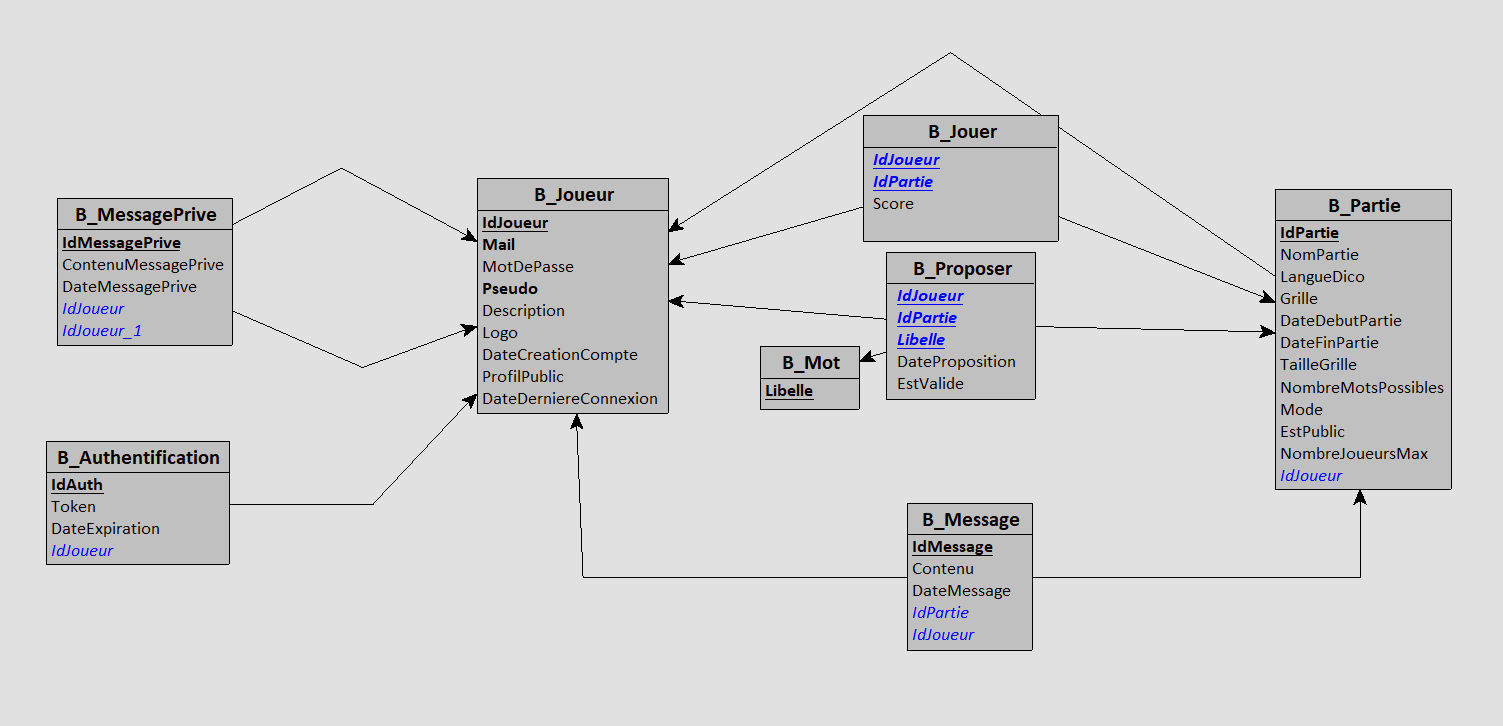
Notre base de données comprend huit tables: B\_Joueur, B\_Partie, B\_Message, B\_MessagePrive, B\_Mot, B\_Jouer, B\_Authentification et B\_Proposer.

Les informations ci-dessous correspondent à notre base de données actuelle. Cette dernière pourra être amenée à recevoir quelques modifications en fonction des difficultés que nous rencontrerons dans le reste du projet.

# Schéma entité-association :

# 

# Schéma relationnel :



***Version textuelle :***

**B\_Joueur** = (*IdJoueur*, Mail, MotDePasse, Pseudo, Description, Logo, DateCreationCompte, ProfilPublic, DateDerniereConnexion);

**B\_Partie** = (*IdPartie*, NomPartie, LangueDico, Grille , DateDebutPartie, DateFinPartie, TailleGrille, NombreMotsPossibles, Mode, EstPublic, NombreJoueursMax #IdJoueur);

**B\_Message** = (*IdMessage*, Contenu, DateMessage, #IdPartie, #IdJoueur);

**B\_Jouer** = (#*IdJoueur*, #*IdPartie*, Score);

**B\_Mot**=(*Libelle*)

**B\_Proposer** = (#*IdJoueur*, #*IdPartie*, #Libelle , DateProposition, EstValide);

**B\_MessagePrive** = (IdMessagePrive, ContenuMessagePrive, DateMessagePrive, #Id\_Joueur, #Id\_Joueur1);

**B\_Authentification** = (IdAuth, Token, DateExpiration, #IdJoueur);

# Dictionnaire de donnée :

**Table B\_Joueur:**

* *IdJoueur*: INT, primary key, auto-increment, l'identifiant unique du joueur
* *Mail*: VARCHAR(320), unique, l'adresse e-mail du joueur
* *MotDePasse*: VARCHAR(64), le mot de passe du joueur crypté en SHA256 (conseillé par la CNIL)
* *Pseudo*: VARCHAR(30), unique, le pseudo choisi par le joueur
* *Description*: VARCHAR(400), une description du joueur
* *Logo*: VARCHAR(64), le hash du logo (<https://fr.gravatar.com/>) qui sera stocké sur le serveur (ex: data/image\_joueurs/HASH.png)
* *DateCreationCompte*: DATETIME, la date de création du compte
* ProfilPublic: TINYINT, indique si le profil du joueur est public ou non
* *DateDerniereConnexion*: DATETIME, la date de la dernière connexion du joueur

**Table B\_Partie:**

* *IdPartie*: INT, primary key, auto-increment, l'identifiant unique de la partie
* *NomPartie:* VARCHAR(50), le nom de la partie
* *LangueDico:* CHAR(3), la langue du dictionnaire utilisé pour la partie
* *Grille*: VARCHAR(200), la grille de mots croisés
* *DateDebutPartie*: DATETIME, la date de début de la partie
* *DateFinPartie*: VARCHAR(50), la date de fin de la partie
* *TailleGrille*: INT, la taille de la grille
* *NombreMotsPossibles*: INT, le nombre de mots possibles dans la grille
* *Mode*: INT, le mode de jeu (exemple : solo, multijoueur)
* *EstPublic*: TINYINT, indique si la partie est publique ou non
* *NombreJoueursMax*: INT, le nombre maximum de joueurs pour cette partie
* *IdChef*: INT, foreign key, NOT NULL, l'identifiant du joueur chef de cette partie

**Table B\_Message:**

* *IdMessage*: INT, primary key, auto-increment, l'identifiant unique du message
* *Contenu*: VARCHAR(200), le contenu du message
* *DateMessage*: DATETIME, la date d'envoi du message
* *IdPartie*: INT, foreign key, NOT NULL, l'identifiant de la partie associée au message
* *IdJoueur*: INT, foreign key, NOT NULL, l'identifiant du joueur qui a envoyé le message

**Table B\_MessagePrive:**

* *IdMessagePrive*: INT, primary key, auto-increment, l'identifiant unique du message privé
* *ContenuMessagePrive*: VARCHAR(200), le contenu du message privé
* *DateMessagePrive*: DATE, la date d'envoi du message privé
* *IdJoueur*: INT, foreign key, NOT NULL, l'identifiant du joueur qui a envoyé le message privé
* *IdJoueur\_1*: INT, foreign key, NOT NULL, l'identifiant du joueur qui a reçu le message privé

**Table B\_Mot :**

* *Libelle*: VARCHAR(200), primary key, le mot proposé

**Table B\_Jouer:**

* *IdJoueur*: INT, foreign key, NOT NULL, l'identifiant du joueur qui joue
* *IdPartie*: INT, foreign key, NOT NULL, l'identifiant de la partie associée
* *Score*: INT, le score du joueur pour cette partie

**Table B\_Proposer:**

* *IdJoueur*: INT, foreign key, NOT NULL, l'identifiant du joueur qui propose le mot
* *IdPartie*: INT, foreign key, NOT NULL, l'identifiant de la partie associée
* *Libelle*: VARCHAR(200), foreign key, NOT NULL, le mot proposé
* *DateProposition*: DATETIME, la date de proposition du mot
* *EstValide*: TINYINT, indique si le mot proposé est valide ou non.

**Table B\_Authentification**

* *IdAuth*: INT, primary key, auto-increment, l'identifiant du token utilisé pour l’authentification
* Token: Varchar (128), token aléatoire généré en SHA256
* *DateExpiration*: DATETIME, la date d’expiration du token
* IdJoueur : INT, foreign key , NOT NULL, l’id du joueur auquel correpond ce token

# Explication de certains choix jugés spéciaux :

Les messages sont séparés en deux tables distinctes. Cela est dû à la différence entre les messages privés qui ne concernent que deux joueurs et les messages envoyés dans le chat d’une partie. Leur différence de nature entraîne une différence de traitement. Il est donc plus simple de les séparer en deux tables. De plus cette différence de nature entraîne une différence au niveau des références à d'autres tables (clef étrangère) ce qui provoquerait un ajout d’informations inutiles : un message privé n’est lié à aucune partie.

La base de données contient beaucoup de dates et de nombre. Comme avec *DateDebutPartie, DateFinPartie, DateDerniereConnexion,...*  Tous ces champs ont été intégrés dans l’optique de faire des statistiques et des graphiques accessibles à tous les joueurs.

Il y a une table que nous savons ne pas être en troisième forme normale, la table B\_Partie. Le NombreDeMotsPossible ne dépend pas uniquement de la clef primaire et devrait en théorie être recalculé en fonction du dictionnaire et non stocké. Cependant le dictionnaire peut changer au cours du temps, la langue est vivante. Et nous avons choisi de ne pas stocker toutes les versions du dictionnaire car cela serait trop lourd. C’est pourquoi nous préférons associer NombreMotsPossibles à l’identifiant de la partie, car il représente la réalité dans les conditions où la partie a été réalisée. (Dans ce cas, le NombreDeMotsPossible n’est pas calculable. C’est pour cela que nous le stockons en mémoire.)

La table B\_Proposer présente à peu près le même cas, en effet nous avons des doutes sur le fait que cette table soit en forme normalisée notamment à cause de DateProposition, EstValide et leurs dépendances avec les autres éléments. Ce modèle nous paraît le plus cohérent, en prenant en considération le problème du dictionnaire qui peut changer au fil du temps.

Concernant le dump,les tables Message et MessagePrive sont sans enregistrement car l’implémentation du chat nous a pris de temps que prévu, nous n’avons donc pas pu enregistrer de données réelles. Néanmoins vous trouverez la structure de ces tables.