

Modèle relationnel :

User (

```
id INT,  
email String,  
login STRING,  
password STRING,  
createAt DATE,  
updateAt DATE,  
listeEmoji STRING,  
coin INT,  
age INT,  
bio STRING,  
score INT,  
profilePicture INT  
)
```

- Le champ *id* constitue la clé primaire de la table.
 - Le champ *email* correspond à l'email entré par l'utilisateur lors de la création du compte. Il s'agit d'un attribut optionnel. En effet cet attribut sert principalement à la récupération du mot de passe en cas de perte ou d'oubli. Il répond à la contrainte d'unicité 1.
 - Le champ *login* correspond au pseudo entré par l'utilisateur lors de la création du compte. Ce champ obéit à la contrainte d'unicité 1 et ne peut être NULL.
 - Le champ *password* correspond au mot de passe entré par l'utilisateur lors de la création du compte et il sera hashé lors de son stockage dans la base de donnée. Ce champ ne peut être NULL .
 - Les champs *createAt* et *updateAt* représente respectivement la date de création du compte et la dernière mise à jour du compte. Ces champs ne peuvent être NULL.
 - Le champ *listeEmoji* représente un tableau d'emoji obtenu par le joueur. Ce champ sera stocké sous forme de STRING dans la base donnée mais sera converti plus tard dans le code en « vrai » tableau d'emoji. Ce champ peut être NULL.
 - Le champ *age* représente l'âge du joueur. Ce champ peut être NULL.
 - Le champ *coin* représente l'argent accumulé par le joueur lorsqu'il gagne une partie. Ce champ ne peut être NULL. Un système d'attribution de coin sera implémenter.
 - Le champ *score* représente le socre du joueur. Ce champ ne peut être NULL. Le classement des joueurs se fera par rapport à ce champ et un système d'attribution de coin sera implémenter.
 - Le champ *bio* représente la description qui sera affiché sur le profil du joueur (s'il décide d'en mettre une bien sur). Ce champ peut être NULL.
 - Le champ *profilePicture* représente la photo de profil du joueur. Il s'agit d'un entier compris entre [1, nbProfilePicture] et sera utilisé par l'équipe du front end. Ce champ peut être NULL.
-

Friend (

```
#isFriendOfLogin  
#hasFriendLogin  
addedAt DATE  
)
```

- Le clé primaire de la table est la clé composé de (isFriendOfLogin, hasFriendLogin)
 - Le champ isFriendOfLogin est une clé étrangère vers la table user et représente l'utilisateur avec lequel on est ami.
 - Le champ hasFriendOfLogin est une clé étrangère vers la table user et l'ami d'un utilisateur.
 - Le champ *addedAt* à la date à laquelle user a accepté l'ami. Ce champ ne peut être NULL.
-

Game (

```
id INT,  
beganAt DATE,  
finishedAt DATE,  
modeSelected STRING,  
numberPlayer INT  
)
```

- Le champ *id* constitue la clé primaire de la table.

- Les champ *beganAt* et *finishedAt* correspondent respectivement à la date de début d'une partie et à la date de fin d'une partie. Ce champ ne peuvent être NULL.
 - Le champ *modeSelected* représente le mode de jeu sélectionner de la partie (CLASSIC'UN ou SPIN'UN). Ce champ ne peut être NULL.
 - Le champ *accessMode* représente le mode d'accessibilité de l'apartie (privé ou public). Ce champ ne peut être NULL.
-

```
UserGame (  
    gameld INT,  
    userLogin INT,  
    score INT,  
    rankInGame INT  
)
```

- Le clé primaire de la table est la clé composé de (*gameld*, *userLogin*).
 - Le champ *gameld* constitue une clé étrangère vers la table Game. Ce champ ne peut être NULL.
 - Le champ *userLogin* constitue une clé étrangère vers la table User. Ce champ ne peut être NULL.
 - Le champ *score* représente le score du joueur à la fin de la partie à la fin de la partie. Ce champ ne peut être NULL.
 - Le champ *rankInGame* représente le rang du joueur à la fin partie. Ce champ ne peut être NULL.
-

Note :

La table UserGame et Friend sont des tables de jointure (nécessaire car relation de type N vers N) entre les tables Game ↔ User et User ↔ User.