

# Rapport du projet de Systèmes de Gestion de Bases de Données : Semestre 7

---

*Projet Jeux de Société*

---

Filière Informatique

Encadrants : S.Lombardy, M.Mosbah

Mathieu LAM

Thibaut LANIER

Léo-Paul MAZIERE

Maël PAUL

Novembre 2022 / Décembre 2022



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Modélisation des données</b>	<b>3</b>
2.1	Contexte de l'application . . . . .	3
<b>3</b>	<b>Schéma relationnel</b>	<b>5</b>
3.1	Contraintes d'intégrité et dépendances fonctionnelles . . . . .	5
<b>4</b>	<b>Implémentation</b>	<b>7</b>
4.1	Créations, suppressions et modifications de nos données . . . . .	7
4.2	Nos requêtes . . . . .	8
<b>5</b>	<b>Mise en forme et utilisation</b>	<b>9</b>

# Chapitre 1

## Introduction

### Présentation du contexte

Ce rapport vise à présenter notre travail réalisé durant le projet de systèmes de gestion de bases de données.

Pendant le travail sur le projet, nous avons pu utiliser à la fois des concepts théoriques sur les bases de données tels que les modèles et schémas entités-associations ou relationnels, mais également travailler le langage SQL en créant notre propre base de données et en la manipulant au travers de diverses requêtes.

Ce projet offrait la possibilité de choisir entre deux sujets, l'un sur un parking et l'autre sur des jeux de société. Nous avons choisi de finalement traiter des jeux car manipuler leurs données nous est apparu comme plus ludique, plus motivant, pour la suite du projet.

# Chapitre 2

## Modélisation des données

### 2.1 Contexte de l'application

L'objectif de ce projet est la création d'une base de données pour des joueurs de jeux de société. Après avoir analysé le sujet, nous en avons ressorti des entités: des joueurs, des jeux, des notes, des thèmes, des catégories, ainsi que des personnes. Ensuite, il a fallu chercher les attributs de chacune de ces entités. Ces différents champs nous étaient fournis par l'énoncé. Un moyen de se rendre compte des différents liens entre elles a été de mettre au point un modèle entités-associations sur ce sujet.

- Les joueurs: Les joueurs ont un identifiant ainsi qu'un pseudo, un prénom, un nom, une adresse mail.
- Les thèmes, les catégories et les personnes: Ces entités nous sont parues évidentes à mettre en place. Dans ces entités, il nous a suffi de mettre un identifiant ainsi qu'un nom, tout en rajoutant un prénom au .
- Les jeux: Les jeux ont besoin d'un identifiant, du nom de leur éditeur, une date de parution, la durée d'une partie, le nombre de joueurs. Si nous avons décidé d'ajouter ces champs très rapidement, il nous a été nécessaire de réfléchir plus longuement sur la manière d'implémenter des extensions. Nous ne voulions pas créer une table supplémentaire "extension", et nous avons pensé à considérer les extensions comme des jeux.
- Enfin, les notes ont été la dernière entité que nous avons créée. Dans cette note ont été ajoutés plusieurs attributs: un identifiant, une note, une date et un nombre de joueurs.

Après avoir décrit toutes les entités que nous voulions utiliser, nous avons également créé des associations entre ces entités. Ci-dessous, vous pouvez observer sur la figure 2.1 notre schéma entités-associations que nous avons présenté avant les vacances. Si les entités et les cardinalités des associations sont les mêmes, lors de notre projet nous l'avons amélioré car plusieurs associations avaient alors le même nom. Dans notre version finale, nous avons renommé l'association "aimer" entre Joueurs et Catégories en "préférer", et changé l'association "avoir" entre Catégories et Jeux en "posséder".

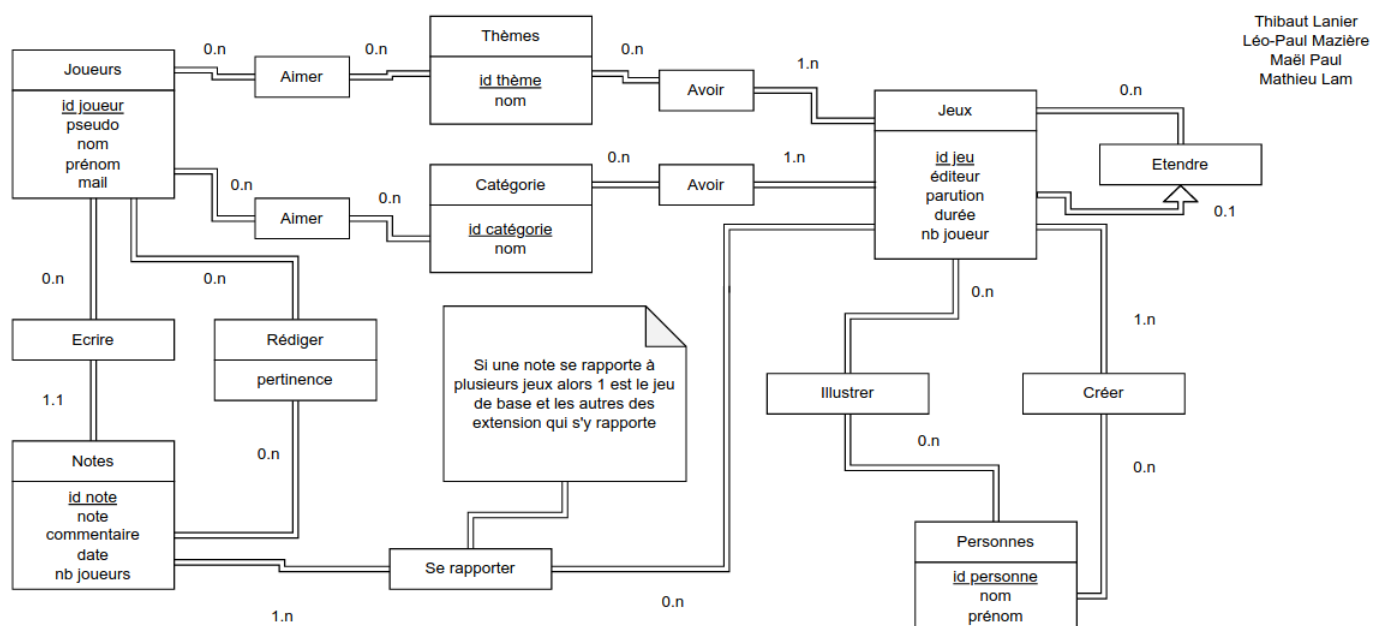


FIG. 2.1 – Schéma entités-associations des joueurs et jeux de société

# Chapitre 3

## Schéma relationnel

### 3.1 Contraintes d'intégrité et dépendances fonctionnelles

Afin de passer du contexte théorique de nos jeux de société à des données plus utilisables, il a été important de passer à un schéma relationnel. Nous avons créé ce schéma relationnel en utilisant les méthodes de passage en schéma relationnel apprises en cours. Nous nous sommes basé sur le schéma relationnel de la figure 3.1 afin de savoir quelles tables créer ensuite pour le passage au langage SQL.

Point important : c'est lors de cette transition qu'on été durablement fixés les deux derniers attributs de notre entité Jeux. Ces attributs permettent la gestion des extensions : le booléen `extension` et la valeur `id_jeu_if_extension`. Le booléen `extension` est vrai s'il s'agit bien d'une extension, dans ce cas le jeu de base auquel elle se rapporte est indiqué via son `id` grâce à `id_jeu_if_extension`.

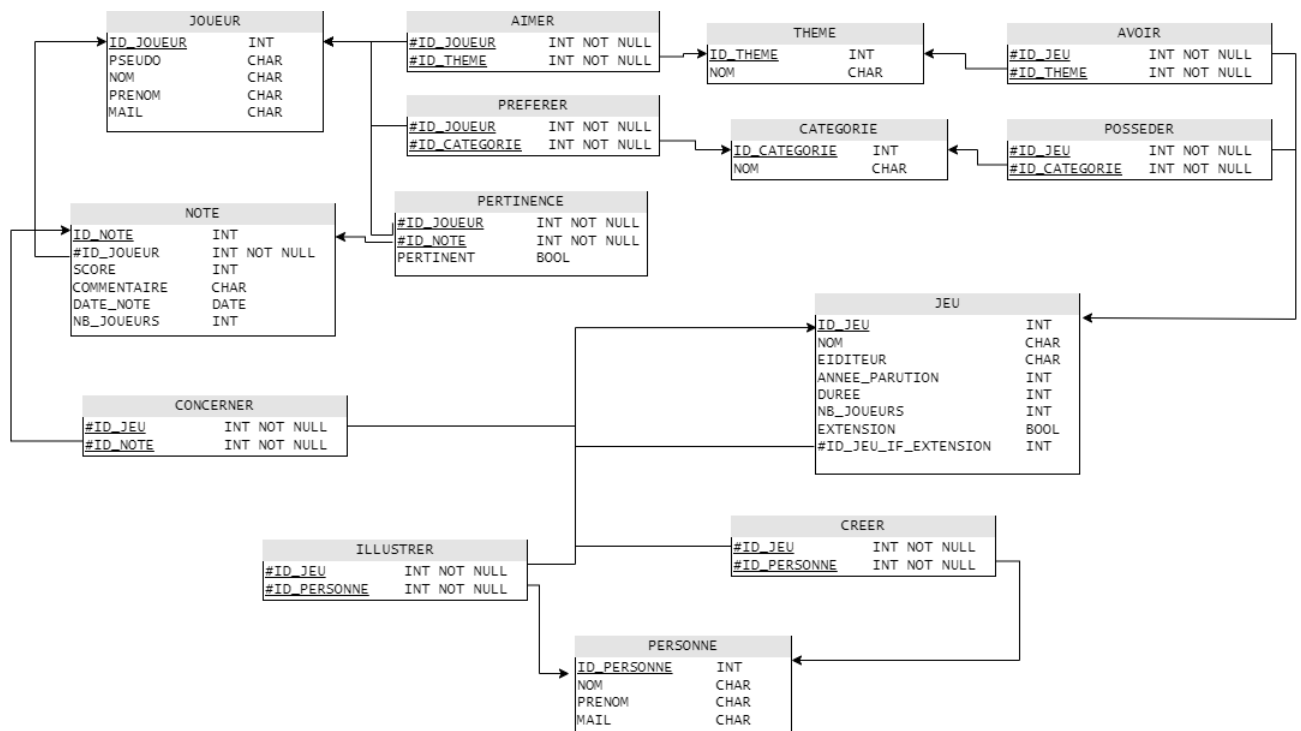


FIG. 3.1 – Schéma relationnel

Sur notre schéma apparaissent notamment des contraintes d'intégrité. Elle imposent à la majorité des clés étrangères, impliquées dans des tables issues d'association du schéma entité-association, à être positives.

# Chapitre 4

## Implémentation

### 4.1 Créations, suppressions et modifications de nos données

Pour mener à bien le projet, nous avons du implémenter notre propre base de données et manipuler cette base de données via des requêtes.

Tout d’abord, nous nous sommes concertés afin de choisir sur quel type de base de données nous allions travailler. Nous avons finalement opté pour PostgreSQL, car elle a été la plus simple à installer et faire fonctionner pour tous les membres du groupe.

Ensuite, il nous a fallu créer notre base de données en créant des tables. Nous avons rassemblé toutes ces créations dans un fichier appelé “create.sql”. Comme expliqué dans la partie précédente, nous sommes partis de notre schéma relationnel afin de savoir quelles tables créer, en reprenant leurs attributs tout en n’oubliant pas de fixer les clés primaires, les clés étrangères ainsi que des contraintes (non nulle, unicité, référence de la clé étrangère). Ces créations de tables ont été faites à partir de la commande “CREATE TABLE”. Après la création de tables, il a fallu ajouter des valeurs dans ces tables. Nous nous sommes demandés préalablement s’il était judicieux de créer un script pour ajouter des données à nos tables, mais finalement nous avons décidé de les ajouter à la main. Nous avons pris cette décision en considérant que cela prendrait plus de temps et n’était pas nécessaire afin d’obtenir une base opérationnelle, et le fait d’ajouter nous mêmes nos éléments nous permettrait un contrôle plus simple de nos fautes.

Ces ajouts dans les tables ont été rassemblés dans le fichier “add\_data.sql” en utilisant simplement la requête “INSERT INTO TABLE”.

Avant de démarrer l’écriture de requêtes, il fallait que nous soyons en mesure de supprimer d’éventuelles valeurs inutiles de nos tables. Cela s’est fait grâce à la commande “DELETE FROM” au début du fichier d’ajout de données sur chacune des tables.

Nous avons du faire attention à l’ordre selon laquelle nous supprimions les valeurs de nos tables, car il fallait respecter les contraintes d’intégrité afin que la base soit être cohérente après chaque opération. C’est pour cette raison que nous avons traité d’abord les tables représentant des associations avant d’enlever celles représentant des entités car les premières dépendaient de ces dernières.

Enfin, pour pouvoir réutiliser notre création de base, il a fallu pouvoir détruire les tables créées. Ces commandes ont été regroupées dans le fichier “clean.sql”, en utilisant la



commande “DROP TABLE” pour chaque en vérifiant avant que la table était existante.

## 4.2 Nos requêtes

De nombreuses requêtes nous ont été données dans le sujet, et nous devions trouver un moyen d’obtenir le résultat de ces requêtes. La plupart des difficultés ont été les mêmes pendant l’élaboration des requêtes: faire attention à utiliser les bonnes jointures, utiliser les bons alias, et faire attention aux requêtes imbriquées. En ayant fait attention à ces différents dangers, 3 grandes familles de requêtes ont été implémentées:

### Consultations

Trois requêtes de consultations nous ont été demandées.

La première requête consistait à classer par mécaniques l’ensemble des jeux critiqués dans un thème donné. Le fichier “classement \_jeu\_bis.2” permet d’avoir le résultat escompté. Pour la liste des commentaires de référant à un jeu dans un thème donné, nous avons compris cette demande de la manière suivante: à partir d’un joueur sélectionné, une catégorie est d’abord choisie au hasard, puis un jeu est sélectionné au hasard parmi cette catégorie. Cette requête se trouve dans le fichier commentaire\_categorie\_prefere\_bis.sql. Enfin, pour un commentaire, il fallait pouvoir connaître la liste des joueurs qui l’ont apprécié. Cette requête est effectuée par le fichier “recup\_info.sql”. La difficulté ici a été de retourner une fonction qui dépende du numéro de note indiqué par l’utilisateur.

### Statistiques

Quatre statistiques de base sur la base de données ont été demandées, ainsi qu’une statistique supplémentaire pondérant les notes par un indice de confiance. Cette dernière requête ainsi que celle classant les notes par indices de confiance ont été longues à mettre en place car elles ont nécessité la création de tables intermédiaires afin d’avoir le résultat final. Afin de pondérer les notes, nous avons utilisé les moyennes des intervalles de confiance.

### Gestion

L’ajout et la suppression nous semblaient assez simples, mais après avoir réalisé des premières versions nous nous sommes rendus compte que cela était plus difficile que prévu. En effet, pour la suppression, enlever une valeur d’une table correspondant à une entité dans notre schéma conceptuel impacte l’ensemble de notre base de données, car les tables représentant les relations de cette entité pourraient également nécessiter des suppressions. Il fallait également pouvoir supprimer des éléments dans le bon ordre, comme dans le cas évoqué précédemment de l’utilisation de la commande DELETE. Le pire cas était celui de la suppression de la table jeu, qui était en relation avec de très nombreuses autres tables. La stratégie employée afin de contourner ce problème a été de créer une table temporaire stockant les identifiants des jeux à supprimer afin de manipuler ces suppressions. De plus, pour les ajouts et les modifications, nous avons un problème car nous avons défini préalablement qu’une extension ne peut pas étendre une autre extension. Nous avons du alors exclure ces cas.

# Chapitre 5

## Mise en forme et utilisation

### Création d'une interface

Si avec les différents fichiers créés, il est possible d'exécuter les différentes requêtes demandées dans le sujet, il nous a semblé judicieux de créer une interface afin de mieux visualiser notre travail et le rendre plus facile d'accès pour un utilisateur.

Dans le sujet, l'utilisation de php pour la création de l'interface était proposé. C'est finalement cette option que nous avons retenu, et si le début a été difficile avec un peu de mal de comprendre l'utilisation de ce langage, après avoir vu les premiers rendus de l'interface nous avons continué dans cette voie.

Afin que notre site communique avec notre navigateur, nous avons utilisé un serveur http, apache2, libre d'accès et simple d'utilisation.

Chaque requête sql que nous avons écrite est reliée à un fichier php permettant l'affichage de ces données.

### Utilisation de notre projet

Il nous a semblé pertinent d'écrire un script, make.sh, permettant de faire rapidement des commandes essentielles et redondantes pendant notre projet, tels que la création de tables, l'ajout d'élément dans ces tables et leur visualisation.

Cependant, l'utilisation la plus facile de notre projet se fait via le site que nous avons créé, accessible directement après l'utilisation de la commande `./make.sh website`. Il est impératif de cliquer sur le bouton "Créer les tables et insérer les données" car comme son nom l'indique, sans ce bouton, il n'y a pas de données dans la base et donc pas de manipulation possible ensuite. Après avoir cliqué sur ce bouton, il est possible d'utiliser les différents onglets en haut à droite de l'écran qui correspondent aux différentes catégories de requêtes que nous avons créé, "mises à jour", "consultations" et "statistiques". La figure 5.1 ci-dessous montre une image de notre site.



FIG. 5.1 – *Page d'accueil du site créé*

D'autres onglets ont été ajoutés afin que l'utilisation soit encore plus simple, avec l'onglet gestion permettant directement d'utiliser les commandes évoquées précédemment de create, add\_data et clean, l'onglet informations permettant de voir le contenu de nos tables, et enfin l'onglet ressources affichant des informations pouvant se révéler utiles telles que le sujet du projet ainsi que le schéma entités-associations et le schéma relationnel.