

Rapport de projet : Base de données relationnelle pour une chaîne de bibliothèques interconnectées

1. Introduction

Contexte du projet

La chaîne de bibliothèques interconnectées souhaite centraliser la gestion des prêts, des collections et des abonnés. Notre objectif principal était de concevoir une base de données relationnelle permettant :

- Le suivi des ouvrages disponibles dans chaque bibliothèque.
- La gestion des abonnés et de leur historique de prêts.
- La réservation et le transfert d'ouvrages entre bibliothèques.
- Le suivi des événements organisés par les bibliothèques.

Équipe

Nous étions un groupe de 4 personnes :

- Yannis, Axel : Responsable de la création des tables SQL et des tests fonctionnels.
- Alban : En charge de la modélisation relationnelle et de la documentation.
- Luka : Normalisation de la base de données, référent du projet et du diagramme relationnel.

2. Déroulement du projet

2.1. Analyse des besoins et reformulation

Lors de la première étape, nous avons identifié et formulé les besoins pour supprimer toute ambiguïté.

Cette étape nous a permis d'établir les fondations de notre schéma relationnel et d'identifier les fonctions critiques à implémenter.

2.2. Modélisation relationnelle

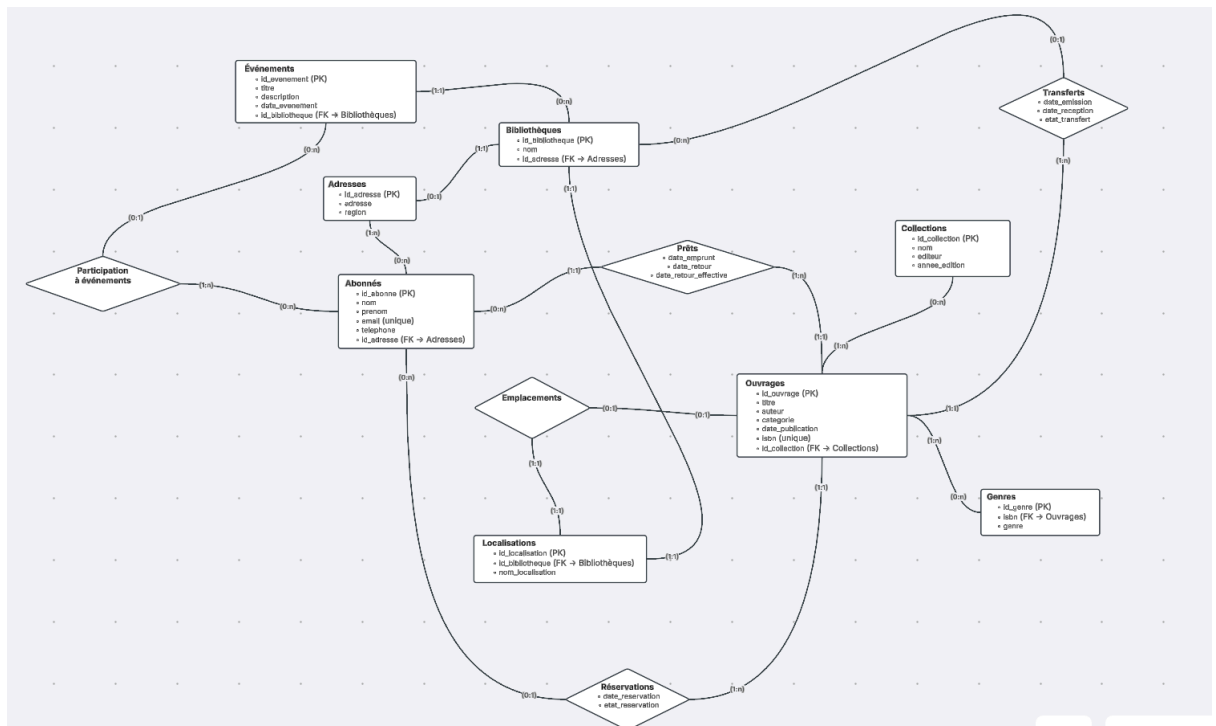
2.2.1. Première tentative (galères avec la normalisation)

Nous avons initialement créé les tables sans passer par un schéma relationnel clair. Rapidement, nous avons constaté des redondances et des incohérences qui nous ont poussés à retravailler la normalisation.

- **Problème** : Certaines tables étaient en 1NF mais pas en 3NF. Les redondances étaient problématiques lors des mises à jour.
- **Solution** : Nous avons repris la modélisation en suivant la méthode vue en cours et recréé un schéma relationnel.

2.2.2. Schéma relationnel final

Nous avons obtenu un schéma relationnel en 3NF :



2.3. Implémentation et tests fonctionnels

Nous avons implémenté le schéma sur **pgAdmin** et créé plusieurs exemples pour tester nos fonctionnalités.

Difficultés rencontrées

- **Trigger manquant** : Nous n'avons pas réussi à implémenter le trigger censé commander automatiquement des livres dès qu'un certain seuil était atteint dans une bibliothèque.
- **Requêtes complexes** : Certaines requêtes imbriquées ont demandé beaucoup d'itérations pour fonctionner correctement.

3. Analyse critique

Points positifs

- Bonne répartition des tâches : chacun a contribué selon ses compétences.
- Flexibilité : Nous avons su nous adapter aux imprévus, notamment lors de la refonte du schéma relationnel.
- Contrainte de la vie réelle : on a essayé de prendre en compte les problématiques de la vie réelle pour avoir une base de donnée le plus réaliste possible.

Difficultés

- Mauvais démarrage : partir directement sur les tables sans schéma a été une erreur qui nous a fait perdre du temps.
- Normalisation : nous n'avons pas réussi à atteindre la BCNF pour certaines relations.
- Triggers : manque de temps pour les implémenter.

Améliorations

- Prévoir plus de temps pour les étapes préparatoires.
- Renforcer nos compétences en SQL avancé (triggers, transactions).

4. Conclusion

Ce projet nous a permis de nous familiariser avec toutes les étapes de la création d'une base de données relationnelle, de l'analyse des besoins à l'implémentation et aux tests. Nous avons acquis des compétences pratiques en normalisation et en conception de requêtes complexes. Malgré les défis rencontrés, nous avons réussi à produire une solution fonctionnelle et évolutive. Si c'était à refaire, nous nous concentrerions davantage sur la planification et la normalisation dès le début.

5. Prochain projet

Pour notre prochain projet, nous allons appliquer les leçons apprises :

- **Planification** : Réaliser une analyse approfondie des besoins et un schéma relationnel dès le départ.
- **Organisation** : Maintenir une répartition claire des tâches et planifier des réunions régulières pour suivre les progrès.
- **Formation continue** : Approfondir nos connaissances en SQL avancé, notamment sur les triggers et les transactions.
- **Gestion des tests** : Prévoir plus de temps pour les tests fonctionnels et l'ajout de cas limites.

Nous avons déjà quelques idées de projet, comme la création d'une base de données pour un système de gestion de cours universitaires ou d'une plateforme de location collaborative. L'objectif sera de consolider nos acquis tout en explorant de nouvelles fonctionnalités.