

Rapport de Projet



Sommaire

Conception	2
S.E.R :	2
Trigger :	6
Transaction	7
Organisation	8
Conclusion	9
REMERCIEMENT	9

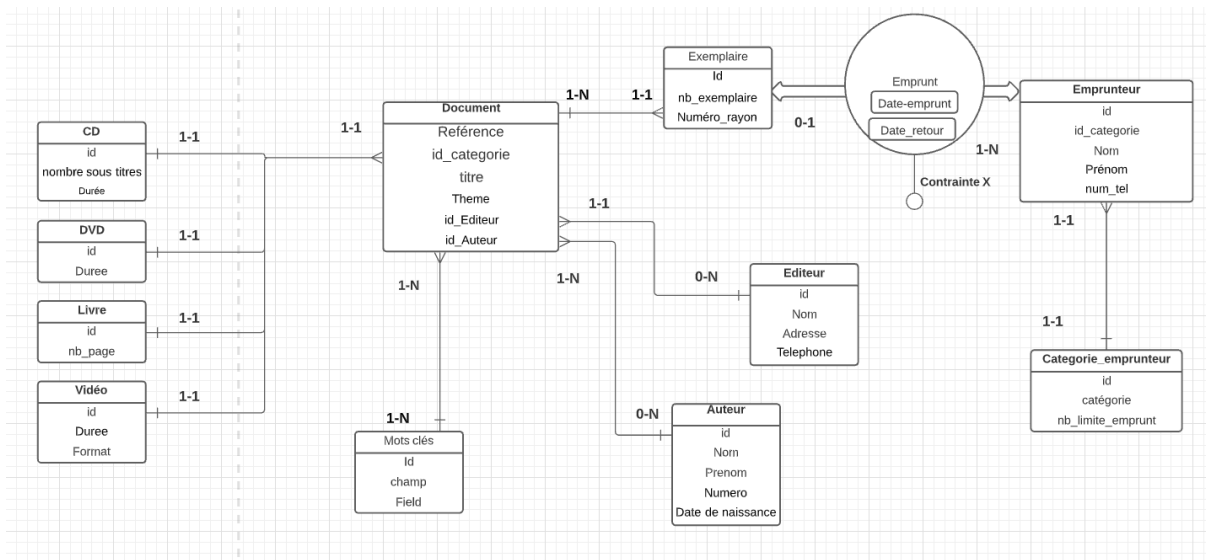
Conception

- SER
- Création des Tables => Modification LMD LAD
- Insertions des données
- Trigger
- Transaction
- Requête

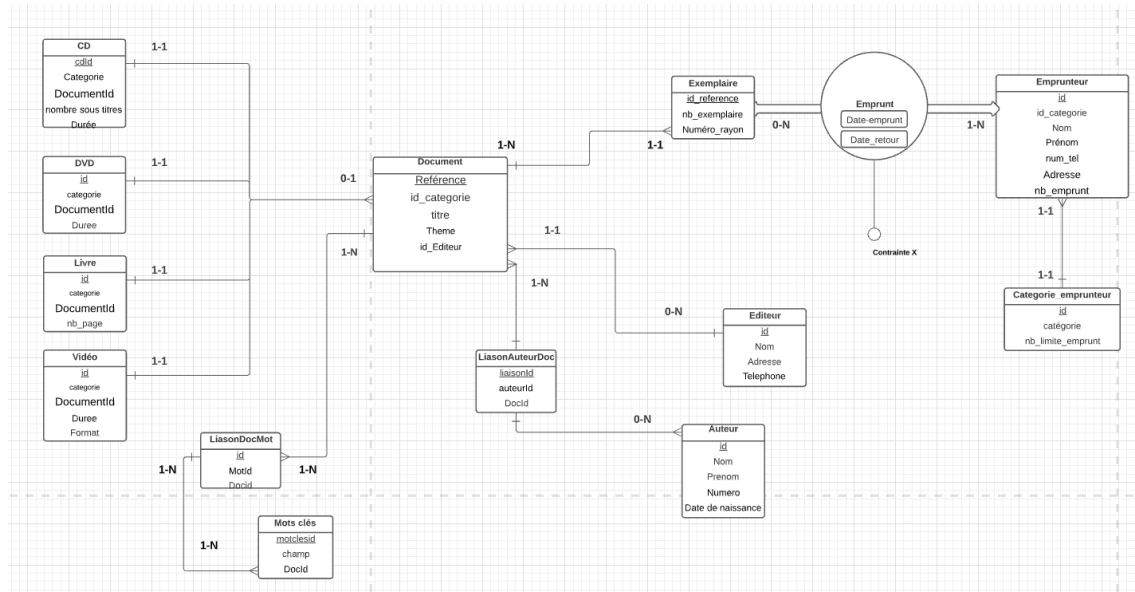
Au cours de la création des tables nous nous sommes rendu compte que nous avions besoin de quelques tables supplémentaires. Une table LiaisonMotDoc nous permet de répertorier les différents mots clés qu'un document peut avoir. Ensuite, une table LiaisonAuteurDoc pour lier les différents auteurs à un document. Nous avons aussi dû faire quelques modifications dans les tables déjà créées pour nous adapter.

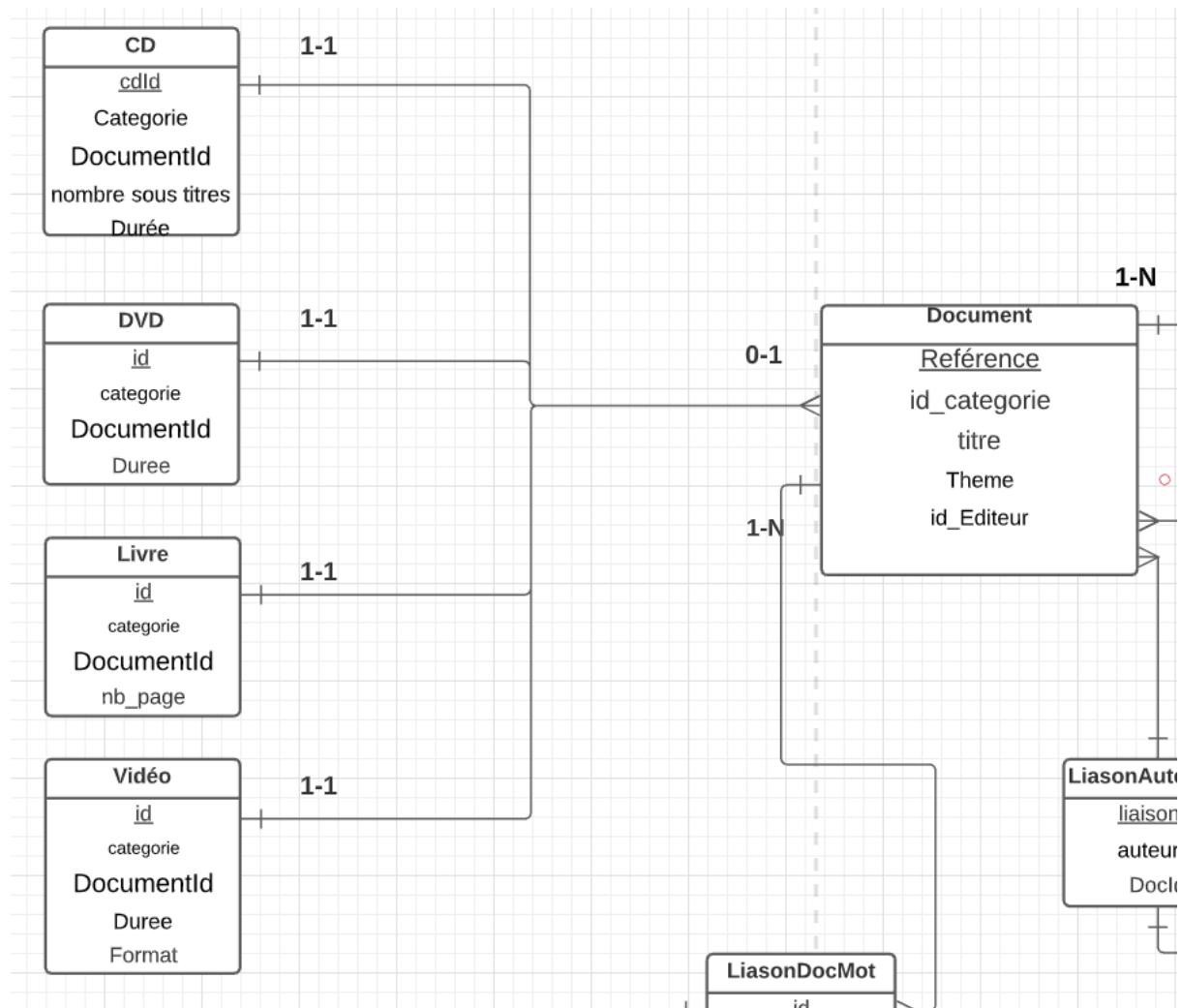
- S.E.R :

SER Version 1.



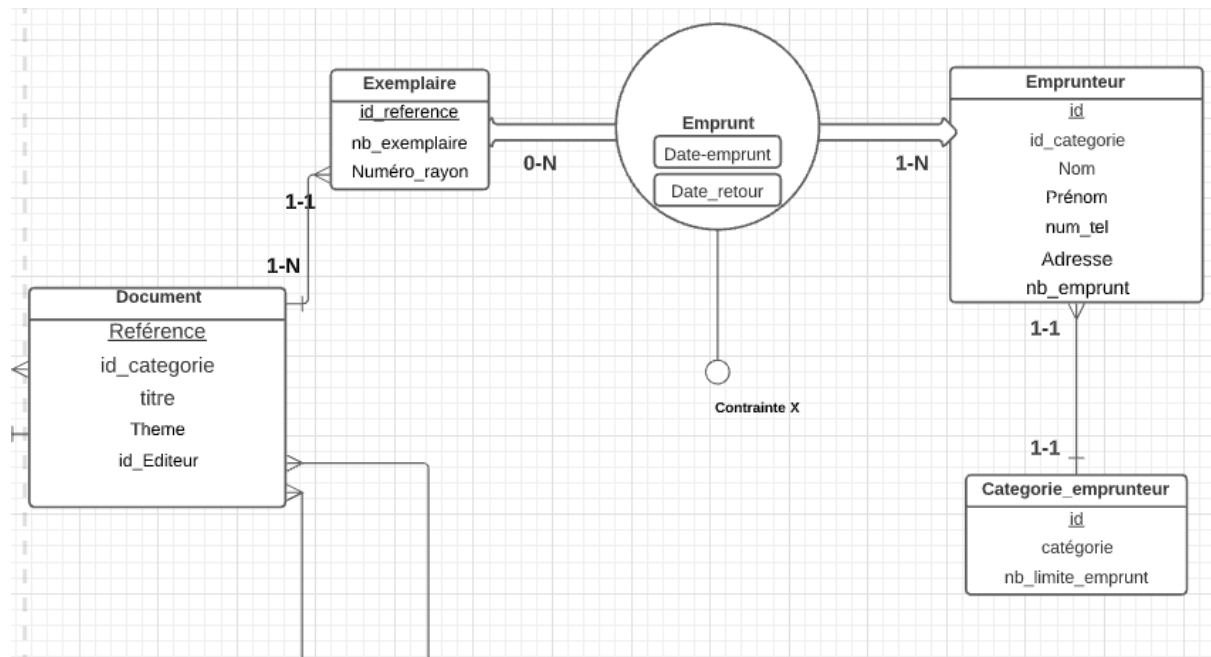
SER Version finale.





Nous avons la table Document qui contient une référence(clé primaire) un id_categorie qui est la clé étrangère qui relie à la catégorie du document, le titre du document, le thème du document et l'id_editeur qui est une clé étrangère correspondant à l'éditeur du document.

Les catégories de document sont représentées par 4 tables, CD, DVD, Livre et Vidéo. Chacune à ses attribut propres mais ils ont tous un id qui correspond à la clé primaire et l'attribut catégorie qui est un int et qui permet de reconnaître la catégorie du document. Les autres attributs sont propres à la catégorie de document en question.



Pour finir nous avons la table Exempleire qui référence un document avec son nombre d'exemplaires de disponible dans la bibliothèque et son numéro de rayon.

la table Emprunteur qui correspond à un emprunteur de la bibliothèque qui peut appartenir à différente catégorie que l'on peut voir avec la table catégorie_emprunteur.

La catégorie emprunteur détermine le nombre d'emprunts maximum possible pour une emprunteur.

Enfin nous avons la table de liaison entre emprunteur et exempleire qui répertorie les emprunts fait pas les emprunteurs avec la date d'emprunt et la date de retour

- Trigger :

Nous avons créé 5 Triggers.

- Le premier est le trigger trig_exempleire qui vérifie avant l'insertion d'un exempleire que le nombre d'exempleire soit bien supérieur à 0 car on ne peut pas insérer un document avec 0 exempleire dans la bibliothèque
- Le suivant est trig_exempleire_update qui vérifie avant une Update de la table exempleire si le nombre d'exempleire après l'update soit bien supérieur ou égale à 0 et donc qu'on ne puisse pas prendre un exempleire si le nombre d'exempleire est égale à 0.
- Nous avons ensuite trig_emprunt_insert qui avant de faire une insertion dans emprunt va update la table emprunteur pour incrémenter le nb_emprunt et la table exempleire pour décrémenter le nombre d'exempleire (ce qui va déclencher les trigger correspondant comme le trig_exempleire_update)

- Le suivant est trig_emprunt_update qui va faire comme le précédent mais va décrémenter le nb_emprunt de emprunteur et incrémenter le nombre_exemplaire de exemplaire qui vont aussi lancé les triggers correspondant pour vérifier qu'il n'y a pas de soucis.
- Le dernier est trig_emprunteur_update qui va vérifier que le nombre de documents empruntés ne dépasse pas le nombre de documents possible à emprunter.

Nous n'avons, par contre, pas réussi à réaliser tous les triggers voulu.

Par exemple, nous n'en avons pas qui vérifient le temps d'emprunt des documents en fonction de leur catégorie. Nous n'avons pas réussi à le faire fonctionner.

Le trigger trig_emprunt_insert et trig_emprunt_update implique que pour l'insertion dans emprunt on ne mette pas de date de retour et que la date de retour ne vienne que via un update. Ce qui semble logique car lorsqu'on emprunte un document on ne sait pas précisément quand on va le rendre.

- Transaction :

Pour les transactions nous en avons fait 6 :

3 pour les ajouts simultanés et 3 pour les suppressions simultanés de documents, emprunts et emprunteurs. Le but ici était de créer des mécanismes permettant de garder une cohérence au sein de notre base. C'est-à-dire de faire en sorte que l'on ne puisse pas ajouter ou supprimer plusieurs fois le même document/emprunt/emprunteur au sein d'une transaction. Pour ce faire, nous ouvrons nos transactions avec des "START TRANSACTION ;" et nous les fermions avec des "COMMIT;". A l'intérieur, nous faisons nos insertions et nos suppressions normalement sauf qu'après un certain moment, on vient mettre des points de sauvegarde "SAVEPOINT;", et afin de pouvoir de revenir à ces versions antérieures, on va utiliser le "ROLLBACK;" qui va nous faire revenir au moment avant le conflit. Au sujet des transactions faites, nous les avons créées dans un premier temps avec la partie d'insertion puis nous les avons aussitôt supprimées avec la partie suppression des transactions.

Organisation

Nathan	Guillaume	Valdrin
Conception du Schéma entité relation		
Trigger / Requête	Requête	Gestion des transactions
création tables Document / Categorie_emprunteur / Emprunt / Emprunteur / Exemplaire / LiaisonAuteurDoc / LiaisonMotDoc	Créations des tables mots clés / DVD /CD / Livre / Vidéo / Éditeur / Auteur	Co-crédation des tables emprunteurs / Documents
Insertion Emprunteur	Insertions des tables mots clés / DVD /CD / Livre / Vidéo / Emprunteur /	Insertion des tables Documents /Emprunt/ Emprunteurs / Auteur/ Liaisons Auteurs Doc

Nous avons commencé par réfléchir et créer le schéma entité relation(SER) en fonction du sujet.

Suite à ça nous avons créé les tables de la base de données en suivant notre SER.

L'étape la plus longue aura été de remplir la base de données avec toutes les entrées mais ça nous aura permis de mettre à jour de problèmes de conception que nous avons corrigés. Par exemple, nous nous sommes rendu compte qu'il fallait plus de tables de liaison que ce que nous pensions au début.

Après que toute la table soit remplie nous nous sommes attelés chacun de notre côté à préparer les Trigger, Requêtes et Transactions.

Conclusion

Pour conclure, nous avons essayé tout le long du projet de respecter le plus possible le cahier des charges ainsi que de respecter le plus possible à notre schéma entité relation (SER) que nous avons modifié à plusieurs reprises afin de le perfectionner/optimiser. Suite à cela nous nous sommes occupés de rendre notre base cohérente afin que les requêtes demandées fonctionnent. Pour que l'on puisse se retrouver dans notre fichier nous avons bien séparé chaque partie afin de pouvoir se focaliser chacun de manière viable sur chaque étape du projet. Nous n'avons pas pu réellement optimiser certaines requêtes possibles car nous avons eu peu d'expérience sur les types d'optimisations (comme les index, tri à fusion ou même le hashage).

REMERCIEMENT

Laurents Fuchs - Enseignant CM et TP/TD

Stéphane Jean - Enseignant TP

Ibrahim Dellal - Enseignant TPS