

Rapport de Projet



Analyse et Interrogation de Bases de Données

Cruchon Joachim

Cougnon Alexandre

Fournier Alexandre



2020 -2021

Université de Poitiers – UFR SFA – Licence

TABLE DES MATIÈRES

SOMMAIRE

Introduction.....	1
I – Conception de la base de données.....	2
1. Schéma Entité Association	2
2. Schéma Relationnel.....	3
II – Création de la base de données	4
III – Transactions de la base de données.....	5
IV – Cohérence de la base de données	6
V – Insertion des données	7
VI – Interrogation de la base de données	8
1. Requête	8
2. Optimisation	8
Conclusion	9
Annexe 1 – Schéma Entité Association (S.E.A) Version 1	10
Annexe 2 – Schéma Entité Association (S.E.A) Version 2	11
Annexe 3 – Schéma Relationnel (S.R) Version 1.....	12
Annexe 4 – Schéma Relationnel (S.R) Version 2.....	13

Introduction

Ce projet de base de données est basé sur la gestion d'une médiathèque. L'objectif est de gérer les analyses et les interrogations des données de cette médiathèque, au sein d'une base de données réalisant certaines transactions, afin de pouvoir connaître chaque document, ainsi que les emprunteurs qui sont des clients de la médiathèque.

Ainsi, dans le cadre du projet de Base de données 2 du semestre 6 de l'université de Poitiers, nous avons réalisé, avec le groupe composé de Cruchon Joachim, Fournier Alexandre et de Cougnon Alexandre, la conception de cette base de données multimédia.

Pour ce faire, nous avons donc réalisé une analyse complète du sujet, à l'aide notamment de certains outils comme le Schéma Entité Associations (S.E.A) ainsi que du Schéma Relationnel (S.R) afin de poser les bases de notre projet, pour par la suite effectuer la création, point par point, de notre base fonctionnelle.

I – Conception de la base de données

Dans le cadre du projet de Base de données 2, nous avons pour objectif de réaliser la gestion des données d'une médiathèque. Pour ce faire, nous avons tout d'abord effectué une analyse conceptuelle du sujet, afin de poser les fondations de notre Bdd, notamment en choisissant les tables ainsi qu'en déterminant leurs associations et leurs relations.

1. Schéma Entité Association

Dans un premier temps, afin de parfaire notre conception, nous avons réalisé le schéma entité de notre base, qui fait office de fondamentaux et de phase préparatoire au schéma relationnel.

En effet, c'est notamment dans celui-ci que se transmet les premières idées et les approches de la conception. Les entités sont posées et c'est au travers des associations formant des phrases que l'on détermine les liens et les cardinalités entre nos tables.

À la suite des idées qui ont émergées de la première analyse du sujet, nous avons réalisé notre premier S.E.A disponible en [Annexe 1](#). En effet, dans cette première conception, nous avons pris en compte les grandes lignes du sujet, et notamment sur les types de documents, ou même sur les catégories d'emprunteur. Or, cette conception est erronée, notamment à cause de quatre erreurs :

- ✗ L'association « `Is_type` » est reliée à document, mais aussi au 4 types de documents. En somme cet erreur n'est pas la plus problématique, car l'erreur réside dans les cardinalités. En effet, la conception du sujet diverge d'une personne à une autre, et ici nous avons choisi dans cette phase de conception de donner une équivalence entre le numéro du document et le numéro de type de document. L'idée n'est pas mauvaise, mais notre façon de manipuler les documents ne colle pas au fonctionnement procédural d'une base de données classique, nous empêchant de procéder comme cela.
- ✗ Le deuxième problème réside ici dans le nombre d'exemplaires par document. En effet, dans le premier S.E.A, il n'était pas pris en compte le fait qu'un document existait en plusieurs fois. Par ailleurs, cet effet sera corrigé dans le deuxième S.E.A avec l'ajout d'une entité dédié à ce fonctionnement.
- ✗ Les cardinalités avec les auteurs et les éditeurs est la troisième faute de cette conception. En effet, et comme indiqué dans le sujet. Un document peut être écrit par plusieurs auteurs, mais pour autant, il est édité par un seul éditeur.
- ✗ La dernière erreur réside dans la façon dont nous avons conçus la table document lors du premier S.E.A. À savoir que le numéro de rayon est dépendant de l'exemplaire du document et non de celui-ci directement.

Toutefois, nous avons réaliser un premier S.R et nous avons débuté les ajouts d'entités et les manipulations de la base dans SQL développeur avant de corriger nos erreurs, que nous avons réalisé plus tard lors de la réalisation de ce projet.

Pour autant, tout au long de notre projet, nous avons modifié en parallèle notre S.E.A afin de réaliser celui qui correspond à la version finale de notre projet ([Annexe 2](#)).

On retrouve notamment dans ce Schéma Entité Association, l'ajout de l'entité « Exemplar » avec le numéro de rayon, qui représente bien un exemplaire de document ainsi que le remaniement de plusieurs cardinalités qui ne correspondaient pas aux attentes du sujet, ni même à une conception réaliste de la médiathèque.

Ainsi, la conception de notre S.E.A a été modifiée à plusieurs reprises tout au long de notre projet. Cependant, nous savons que cette partie de la conception ne peut pas être parfaite dès le premier essai, pour autant cela reste un passage obligatoire afin de pouvoir poser les fondations des entités et des associations de la base.

2. Schéma Relationnel

Afin de donner suite à la conception de la base de données, nous avons réalisé le schéma relationnel de notre base.

Étape forcément indispensable au bon réalisme d'un projet de BD, le SR a pour objectif de poser les relations entre les tables, et notamment d'apporter les champs manquants à chaque entité, notamment les clés étrangères, qui sont traduites par les associations du S.E.A. Ainsi, le S.R. est une étape à faire à postérieure du S.E.A et est l'ultime phase de la conception avant la partie SQL du projet.

De plus, c'est à l'aide du logiciel Access de la série Office que nous avons réalisé notre S.R déduit de notre premier S.E.A ([Annexe 3](#)). Comme vu précédemment avec la première version du S.E.A, la conception de ce Schéma Relationnel concentre quelques problèmes majeurs, notamment corrigés après coup lors de la programmation en SQL au sein de SQL développeur.

Par ailleurs, comme vu précédemment, le S.R a donc vu plusieurs évolutions et améliorations tout au long de ce projet, afin d'obtenir la version courante du SR disponible en [Annexe 4](#).

Ainsi, nous sommes passés par une longue phase de conception afin de réaliser notre projet, notamment car le sujet porte à plusieurs conceptions et visualisations différentes, et qu'il est bon de fixer les objets et d'imager un maximum ces idées, que ce soit au travers du S.E.A, du S.R, ou même de certaines contraintes d'intégrités afin de pouvoir poser les fondations de la base de données.

II – Création de la base de données

Dès lors que les bases de notre conception ont été posées, nous nous sommes lancés sur la réalisation de notre code en SQL. En effet, c'est avec le logiciel SQL développeur que nous avons réalisé nos tables.

La création des tables en SQL reste simple, il suffit d'appeler un CREATE TABLE, suivi de l'appel des champs avec leurs types ainsi que de spécifier quels sont les clés primaires et étrangères et la table est réalisée selon les conditions du S.R.

Exemple :

```
CREATE TABLE Documents (  
  Id_documents INT,  
  Title VARCHAR2(255),  
  Theme VARCHAR2(255),  
  Editor VARCHAR2(255),  
  CONSTRAINT PK_Documents PRIMARY KEY (Id_documents),  
  CONSTRAINT FK_Documents_Editor FOREIGN KEY(Editor) REFERENCES  
  Editors(Name_Editors));
```

III – Transactions de la base de données

Nous avons par la suite, ajouté quelques données à la base manuellement. Tout d’abord, en désactivant l’AUTOCOMMIT qui est activé automatiquement et commit toutes les requêtes de type INSERT DELETE UPDATE. On peut ensuite faire l’ajout et commit manuellement.

```
INSERT INTO EXEMPLAR(ID_EXEMPLAR, SHELF, DOCUMENTS)
VALUES(0, 102, 3);
commit;
```

Dans un second temps, nous avons utilisé la gestion de transaction pour gérer certains cas. La première transaction gérée est l’ajout d’un document, nous avons choisi ici de ne pas avoir plusieurs documents ayant le même nom, ceci de manière arbitraire.

La transaction se déroule ainsi :

- On verrouille la table DOCUMENTS en écriture : LOCK TABLE DOCUMENTS IN EXCLUSIVE MODE.
- On ajoute un point de sauvegarde au début de la transaction.
- On fait notre ajout et on vérifie combien de titre sont similaire dans notre base, s’il y en a plus de 1 alors on revient au point de sauvegarde grâce au ROLLBACK.

De la même façon, nous avons également ajouté une condition pour l’ajout d’un emprunteur qui est d’avoir un nom et prénom différent d’un autre emprunteur.

On retrouve les mêmes étapes que pour la première transaction :

- Verrouillage de la table
- Ajout d’un point de restauration
- Ajout dans la base et vérification d’un doublon nom/prénom, auquel cas retour au point de restauration avec le ROLLBACK.

IV – Cohérence de la base de données

Nous avons poursuivi notre projet en ajoutant dans triggers afin vérifier les différentes incohérences dans notre base.

Il y a deux types de triggers dans notre projet, dans un premier temps nous avons ajouté des triggers qui pour chaque table ayant un id comme clé primaire, incrémente cet id lors d'un ajout. Ces triggers nous permettent de ne pas nous occuper nous-mêmes de l'incrémentation des clés primaire et de ne pas avoir de conflit sur celle-ci.

Dans un second temps nous avons ajouté dans triggers qui vérifie certaines propriétés lors d'un ajout, une modification ou une suppression dans notre base. Les triggers sont les suivants :

- Vérification que la date de naissance d'un auteur est inférieure à la date du jour : comparaison de la date entrée avec sysdate.
- Vérification que le nombre de pages d'un livre n'est pas inférieur à 0.
- Lors de l'ajout d'un emprunt, il faut que la date de fin d'emprunt soit plus récente que la date de début.
- Pour qu'un document soit emprunté, il faut qu'il soit disponible, pour ça on vérifie s'il n'y a pas un emprunt en cours avec ce document où la date de fin est dans le futur.
- Afin d'emprunter un document, l'emprunteur ne doit pas avoir dépassé son nombre de documents maximum pouvant être emprunté en même temps. On regarde en fonction de sa catégorie le nombre de documents qu'il peut emprunter, si ce nombre est déjà atteint alors le trigger se déclenche.
- Avant la suppression d'un emprunt dans la base, il faut vérifier que cet emprunt est bien terminé.
- Pour la suppression d'un client dans la base, il faut vérifier qu'il n'a pas d'emprunt en cours.
- De même pour un exemplaire de la base, si l'on souhaite le modifier il faut s'assurer qu'il n'est pas en cours d'emprunt.

V – Insertion des données

À la suite de l'ajout des triggers, et afin de pouvoir manipuler les données de notre base avec des requêtes, nous avons inséré des données dans chacune de nos tables.

De plus, l'insertion de données prend en compte les clés étrangères et notamment les types entre la clé primaire et la clé étrangère correspondante, ainsi afin de procéder à des insertions fonctionnelles, nous avons effectué plusieurs vérifications de types afin de vérifier leurs compatibilités et surtout si ceux-ci correspondent avec ce qui est attendu dans le sujet.

Pour ce qui est de l'insertion en SQL, il faut appeler INSERT INTO avec tous les champs présents dans la table afin de créer une ligne de l'entité.

Exemple :

```
INSERT INTO EDITORS (NAME_EDITORS , ADRESSE , PHONE_NUMBER)  
VALUES ('DNC media', '93-2 Myeong Dong SEOUL', '02 333 2514');
```

VI – Interrogation de la base de données

1. Requête

Afin d'effectuer la partie analyse de la base de données, nous avons réalisé une série d'interrogations sous forme de requête SQL avec SQL développer en accord avec celles imposées par le sujet.

Beaucoup de requêtes appellent des conditions de jointure et de liens entre les tables, impliquant une bonne réalisation de la base au préalable comme vu dans les parties [I](#) et [II](#).

2. Optimisation

Pour donner suite à l'écriture de nos requêtes, nous avons réfléchi à quels seraient les index qui correspondraient à la meilleure optimisation que nous avons traduit sous forme d'un tableau :

<u>Index</u>			
Arbre	Bitmap	Hachage	
		Statique	Dynamique
2,3,6,7	1,8,10,15	4	5,9,17,18,19,20

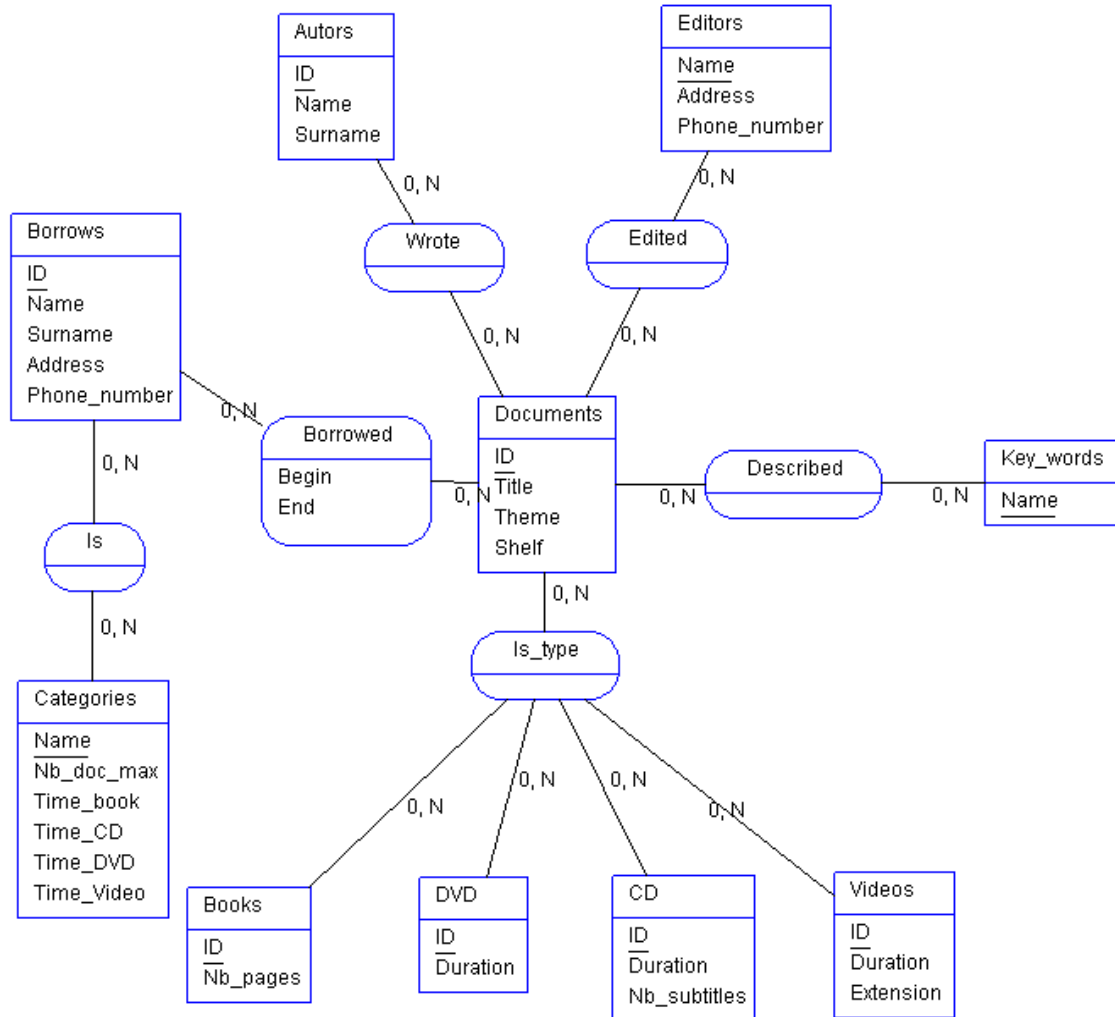
Après réflexion, les requêtes 11, 12, 13, 14, 16 ne nécessitent pas d'index pour l'optimisation.

Conclusion

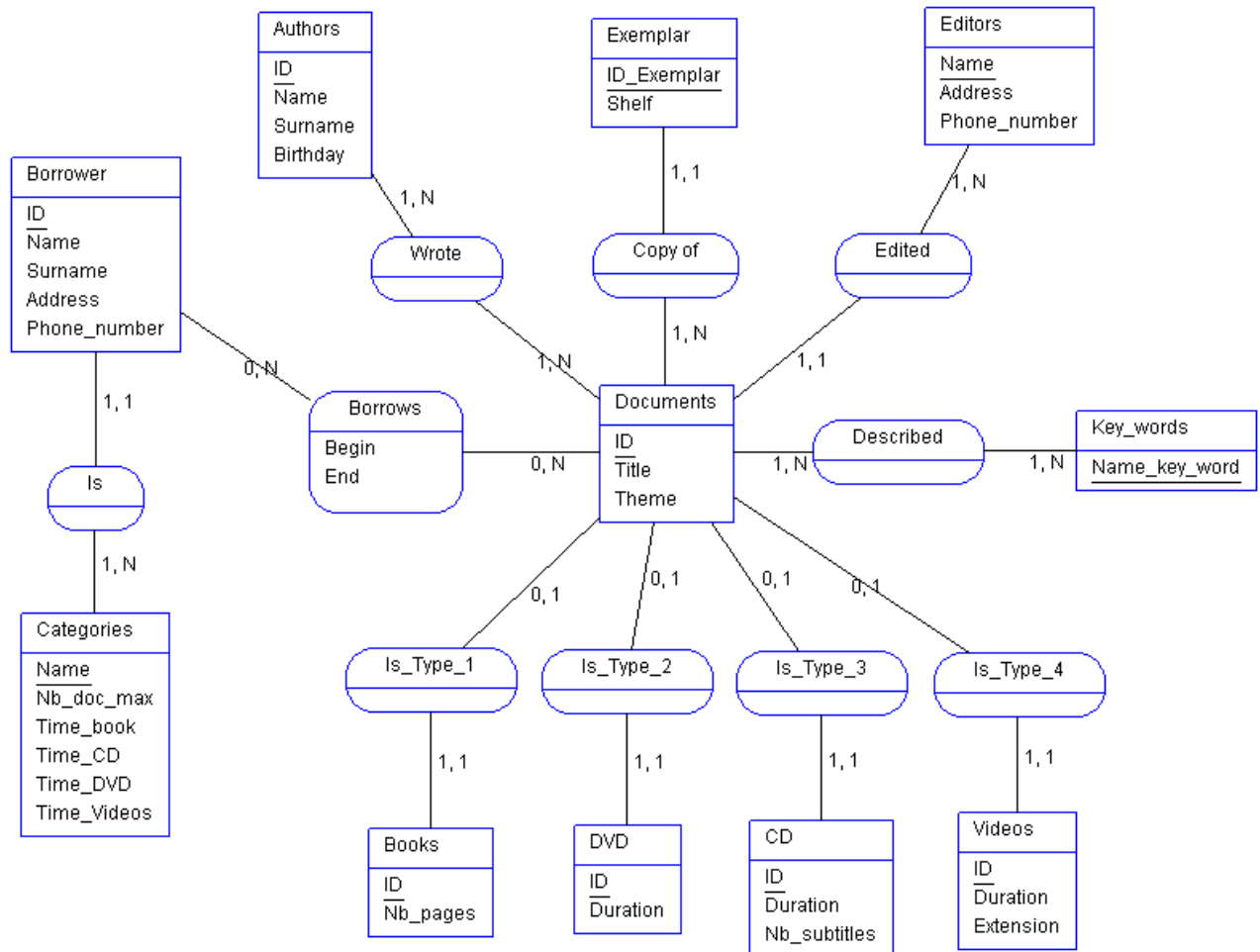
Pour conclure, ce projet nous a permis d'exploiter les ressources vues en Travaux Pratiques dans un cas concret de base de données d'une médiathèque.

En effet, tout au long de la réalisation de celui-ci, nous avons tout d'abord effectué une phase de conception schématisée afin d'ordonner nos idées des premières lectures du sujet. Suivi de la création des tables et de la manipulation des données, ce projet nous a permis d'aborder et de répartir à plusieurs, le travail d'analyse et d'interrogation d'une base de données que nous pouvons, à l'avenir, manipuler à nouveau.

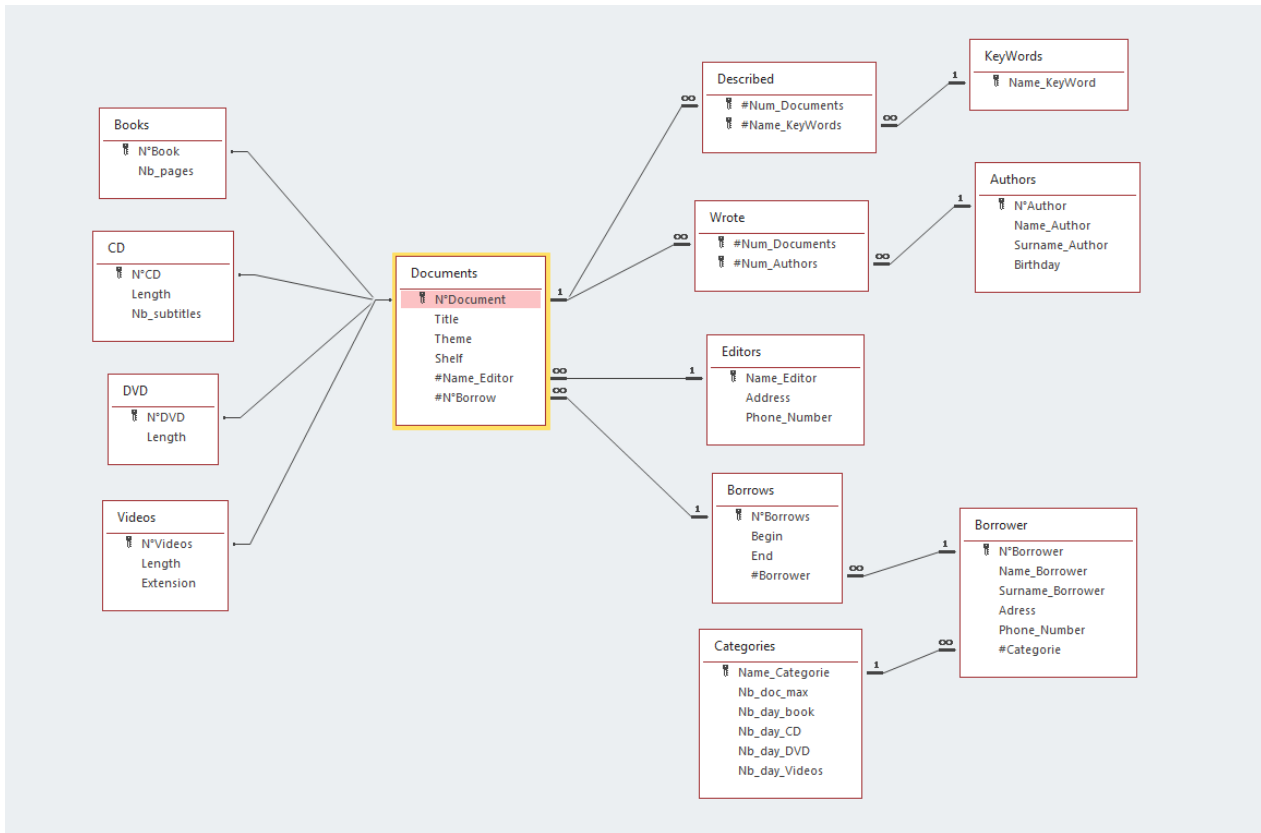
Annexe 1 – Schéma Entité Association (S.E.A) Version 1



Annexe 2 – Schéma Entité Association (S.E.A) Version 2



Annexe 3 – Schéma Relationnel (S.R) Version 1



Annexe 4 – Schéma Relationnel (S.R) Version 2

