Silly Luca

Loïc David

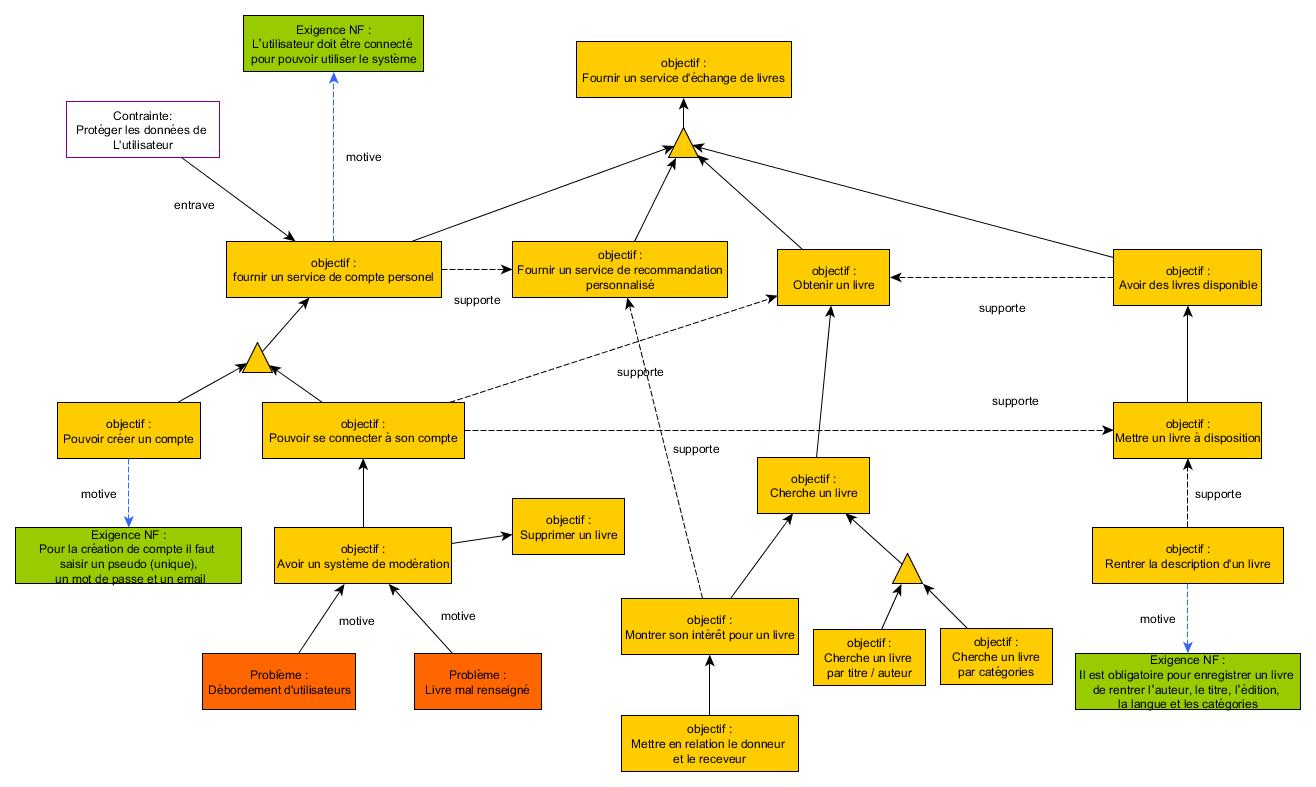
**Rapport CheckPoint 1**

1. **Introduction**

Après avoir lu un livre, il est fréquemment condamné à prendre la poussière sur une étagère. En effet peu de gens relise leurs livres plusieurs fois alors que d’autre n’ont pas forcément la capacité à acheter des livres neufs régulièrement

Notre projet aurait pour but de permettre aux gens de donner des livres qui ne les intéressent plus à une autre personne qui lui souhaiterait les avoir. Pour cela nous allons développer un système permettant aux utilisateurs de donner un livre avec sa description (titre, auteur, date, catégorie, …), trouver un livre parmi tous ceux disponibles et de montrer son intérêt. Il serait aussi intéressant de proposer à l’utilisateur des livres en fonction de ce qui l’intéresse.

1. **Analyse des objectifs**

****

Objectif :

* Mettre un livre à disposition :
  + Un utilisateur enregistré peut mettre à disposition un livre qu’il possède et qu’il souhaite donner
  + Pouvoir rentrer des informations sur le livre : titre, auteur, \*état, Edition, \*date, catégories (possiblement des « cases » à cocher), langue, \*pages, \*résumer, \*isbn, \*autre

(\*facultatif)

* Chercher des livres
  + On peut chercher un livre par titre, auteur, catégories
* Montrer son intérêt pour un livre
  + Pour faire cela l’utilisateur doit être connecter
  + Après avoir fait une recherche, on peut sélectionner un livre et cliquer sur une « case » pour le mettre en intérêt
* Mettre en place un système de recommandation
  + Page d’accueil propose des recommandations
  + Ces recommandations sont en fonctions des livres mis en intérêt par l’utilisateur (auteur, catégories en commun) basé sur les 20 derniers livres
* Profil
  + L’utilisateur doit avant toute chose créer un compte ou se connecter pour pouvoir utiliser notre système
  + Création de compte : pseudo (unique), mot de passe, email,
* Mise en relation des tiers :
  + Une fois qu’un utilisateur porte intérêt à un livre on signale au propriétaire la mise en intérêt de son livre avec un mail contenant le pseudo, l’adresse mail et la note de l’utilisateur
  + On considéra que la transaction est faite lorsque l’ancien propriétaire retire le livre et on retira le livre de la base.

Exigences non fonctionnelles :

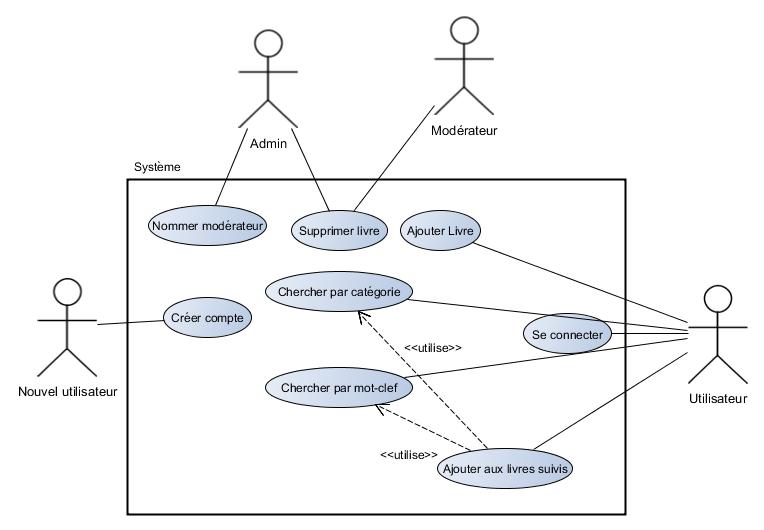
* Il est obligatoire pour enregistrer un livre de rentrer l’auteur, le titre, l’édition, la langue et les catégories ;
* Le système doit être disponible 24 heures sur 24 ;
* L’utilisateur doit être connecté pour pouvoir utiliser le système ;
* Ces recommandations sont en fonctions des livres mis en intérêt par l’utilisateur (auteur, catégories en commun) basé sur les 20 derniers livres ;
* Pour la création de compte il faut saisir un pseudo (unique), un mot de passe et un email ;
* Protéger les données des utilisateurs ;
* Le système est en français ;

1. **Use Case**

Nos use case représentent la manière dont les utilisateurs interagiront avec notre système. Pour cela nous avons utilisés 4 acteurs :

* Un nouvel utilisateur représente une personne qui n’a jamais utilisé notre système ;
* Un utilisateur représente une personne qui a déjà utilisé notre système et donc a déjà un compte ;
* Un admin est un utilisateur avec toutes les permissions possibles par notre système
* Un modérateur est un utilisateur qui possède certaines permissions qu’un utilisateur ne possède pas.

Voici ci-dessus le schéma correspondant à nos use case :

****

Les scénarios correspondant à nos Use Case sont les suivants :

* Créer un compte :

L’utilisateur saisi un pseudo qui l’identifiera dans la base de données, il saisit ensuite une adresse mail, servant à le mettre en relation avec les autres utilisateurs.

L’utilisateur doit ensuite saisir 2 fois de suite le mot de passe qu’il souhaite utiliser.

Si tous les champs sont remplis et qu’aucun utilisateur n’utilise le pseudo et/ou l’adresse mail, le compte est créé.

* Se connecter :

L’utilisateur saisit son pseudo et son mot de passe, si ce couple pseudo/mot de passe est reconnu, l’utilisateur est connecté.

* Ajouter Livre :

L’utilisateur doit saisir le titre du livre, l’auteur, l’état du livre, la langue, la/les catégorie/s, et l’édition.

L’utilisateur peut aussi renseigner le code ISBN, un résumé, la date de parution, le nombre de page.

* Chercher par mot-clef :

L’utilisateur tape un ou plusieurs mots-clefs dans la barre de recherche.

Les mots-clefs saisis sont comparés aux titres des livres disponibles et aux noms des écrivains dont certains livres sont disponibles.

* Chercher par catégorie :

L’utilisateur coche les catégories qu’il souhaite.

Le résultat de la recherche sera tous les livres contenant une ou plusieurs catégories saisies.

* Ajouter aux livres suivis :

Après avoir fait une recherche l’utilisateur peut signaler son intérêt pour un livre.

Le système envoie alors un courriel au propriétaire du livre pour lui signaler l’intérêt d’un utilisateur pour son livre et lui fournit l’adresse mail de la personne pour qu’ils puissent communiquer.

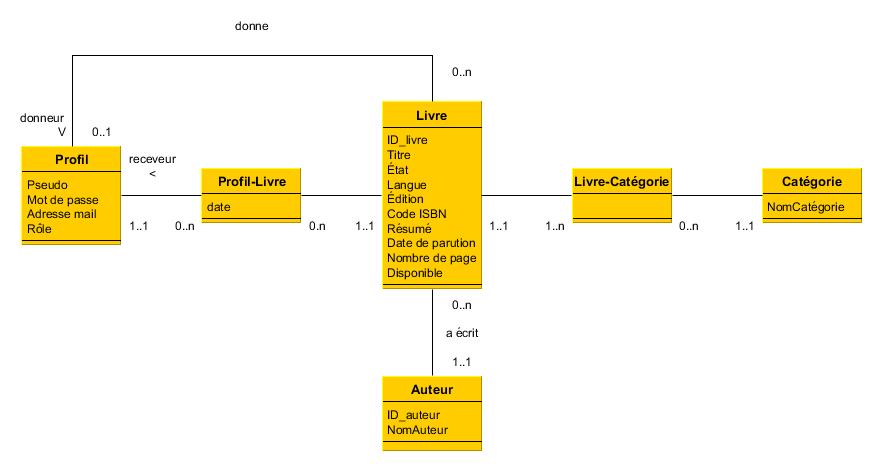
* Supprimer livre :

Les admins et les modérateurs peuvent supprimer des livres, qu’ils/elles jugent comme étant mal définis ou inappropriés.

* Nommer modérateur :

Les admins peuvent nommer d’autres personnes comme modérateur, leur permettant ainsi d’aider à gérer le site.

1. **Schéma conceptuel**

****

Le schéma conceptuel ci-dessus permets de visualiser la fonctionnement notre futur base de données.

Nous avons dans notre schéma 4 classes et 2 classes associations.

**Classes :**

* Profil : Cette classe permet de représenter l’utilisateur avec 4 attributs dans le système.
  + Pseudo correspond au nom de l’utilisateur et il est unique, utile à la connexion ;
  + Mot de passe correspond au mot de passe de l’utilisateur, utile à la connexion ;
  + Adresse mail correspond à l’adresse mail de l’utilisateur, elle permettra de mettre en relation le donneur et le demandeur ;
  + Rôle permet d’identifier la permission d’un utilisateur, par défaut le rôle est NULL pour un utilisateur lambda, il peut aussi correspondre à ADMIN ou MODO selon les permissions désirées.
* Livre : Cette classe permet de représenter à l’aide de 10 attributs les livres que les utilisateurs veulent mettre à disposition
  + ID\_livre permet d’identifier un livre par rapport à un autre sans causer de problème avec des livres ayant le même titre (attribut unique) ;
  + Titre permet de définir le nom du livre correspondant à son ID ;
  + Etat est un attribut qui par défaut prend une valeur nul et permet au donneur de décrire dans quel état se trouve le livre qu’il souhaite donner ;
  + Langue permet de préciser dans quelle langue le livre est écris ;
  + Édition correspond à l’édition du livre en question ;
  + Code ISBN est un attribut qui par défaut prend une valeur nul et elle correspond au numéro ISBN du livre ;
  + Résumé est un attribut qui par défaut prend une valeur nul et elle correspond à une description du livre ;
  + Date de parution est un attribut qui par défaut prend une valeur nul et elle permet de définir la date à laquelle le livre est sorti ;
  + Nombre de page est un attribut qui par défaut prend une valeur nul et elle permet de définir le nombre de page du livre ;
  + Disponible est un attribut Boolean qui prend par défaut la valeur 1 et permet de définir l’état d’échange du livre, 1 correspond à encore disponible et 0 correspond à plus disponible.
* Catégorie : Cette classe permet d’avoir une liste de tous les genres de livres possibles, ils sont uniques.
* Auteur : Cette classe permet d’avoir une liste d’auteurs en associant un id et un nom pour éviter de problèmes avec des auteurs qui auraient le même nom.

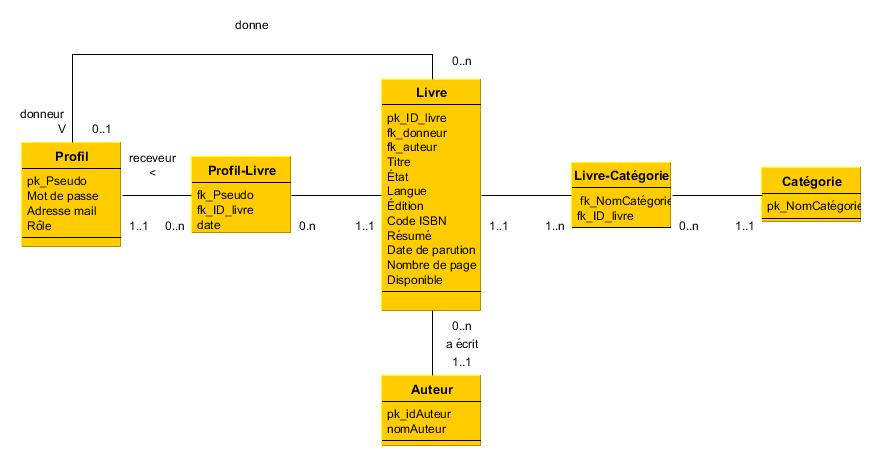
**Classes associations :**

Une classe association est une classe qui permet d’éviter une relation de plusieurs à plusieurs entre 2 classes. Dans notre système nous en avons deux :

* Profil-Livre : Cette classe association permet de faire le lien entre un livre et la personne intéressée. On a aussi rajouté un attribut date permettant de définir dans le temps l’étape de mise en intérêt pour faciliter le système de recommandation.
* Livre-catégorie : Cette classe association permet d’associer un livre avec une ou plusieurs catégories

1. **Schéma logique / relationnel**

A l’aide du schéma conceptuel et des règles d’intégrités nous avons pu construire le schéma relationnel ci-dessous nous permettant d’implémenter notre base de données

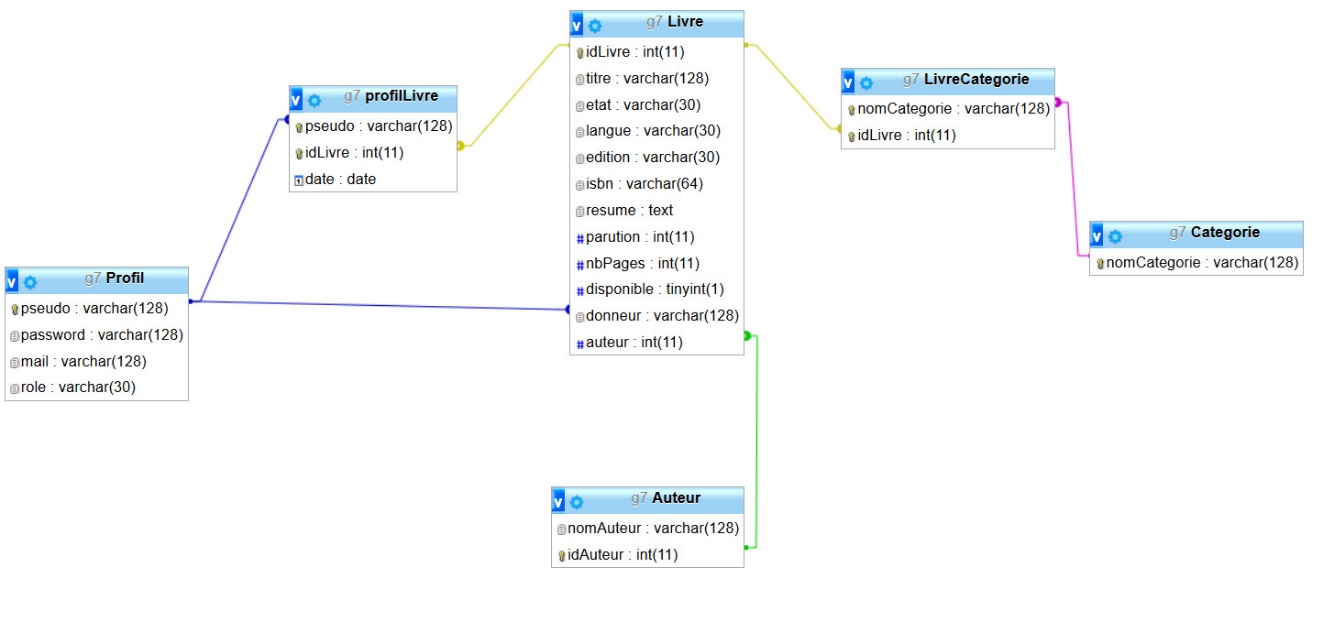


Les attributs précédés par « pk\_ » correspondent aux clés privées qui permettent de définir un objet unique dans la base de données.

Les attributs précédés par « fk\_ » correspondent aux clés étrangères permettant de lier les objets.

Nous avons ensuite utilisé PhpMyAdmin pour implémenter notre base de données, nous avons un total de 6 tables : Auteur, Categorie, Livre, LivreCategorie, Profil, ProfilLivre.

Après l’implémentation nous avons obtenu à l’aide de l’outil un autre schéma relationnel :



Dans la table profilLivre nous avons :

* L’attribut « pseudo », lié à la table profil correspondant à la personne voulant un livre
* L’attribut « idLivre », lié à la table Livre correspondant au livre voulu

À l’aide de ces 2 attributs nous pouvons facilement retrouver qui sont les personnes qui veulent un livre ou quels livres veut une personne.

Dans la table Livre :

* L’attribut « donneur » correspond à la clé primaire de la table profil qui a mis le livre à disposition
* L’attribut « auteur » correspond à la clé primaire de la table Auteur permettant d’éviter le problème de fautes de nom ou d’auteurs qui ont le même nom

Dans la table LivreCategorie :

* L’attribut « nomCategorie », lié à la table Categorie correspondant aux catégories d’un livre
* L’attribut « idLivre », lié à la table Livre correspondant au livre en question

À l’aide de ces 2 attributs nous pouvons associer un livre avec toutes ses catégories car souvent un livre à plus d’une catégorie.

1. **Données de test**

Nous avons rempli nos tables avec quelques données afin de pouvoir vérifier son bon fonctionnement et nous en avons profité pour réaliser quelques requêtes SQL qui seront probablement utilisés dans le futur.

Pour remplir la table Profil nous avons utilisés deux générateurs dont le premier nous permettant de générer un pseudo et une adresse mail : <https://www.json-generator.com/> avec ce format Json :

[

'{{repeat(5, 7)}}',

{

pseudo: '{{firstName()}} {{surname()}}',

mail: '{{email()}}',

password : '{{""}}'

} ]

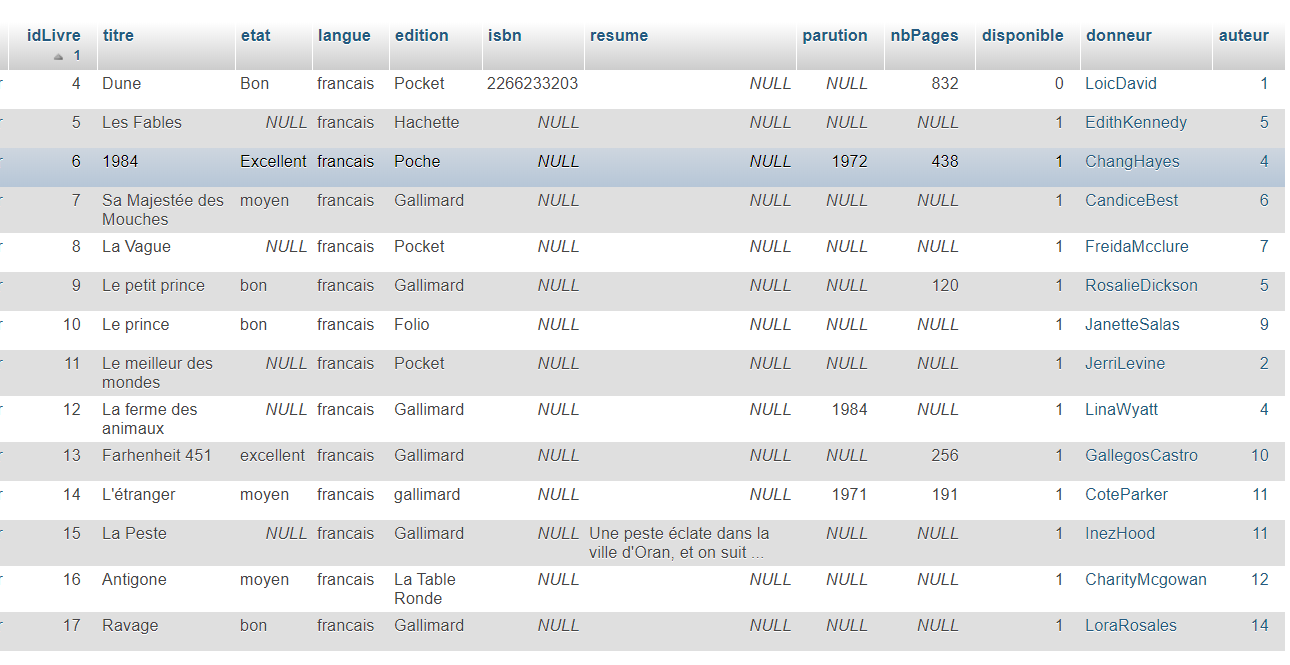
Ensuite pour générer un mot de passe nous avons utilisé ce site : <https://www.lastpass.com/fr/password-generator>

Ensuite, étant donné le peu de données que l’on a remplis pour l’instant nous n’avons utilisés que nos connaissances pour remplir les autres tables (livre, auteur…).

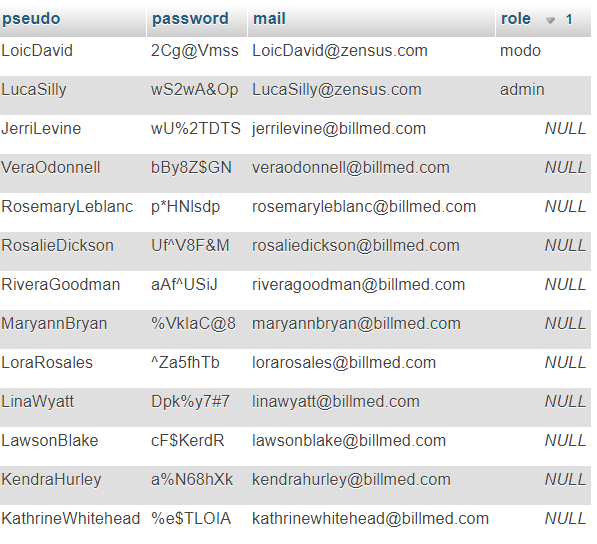
La base de livres s’étoffera lors de l’avancement du projet avec tous les tests que l’on exécutera lors du développement du service web l’utilisant.

Voici un exemple de tables remplies :

Pour la table livre :

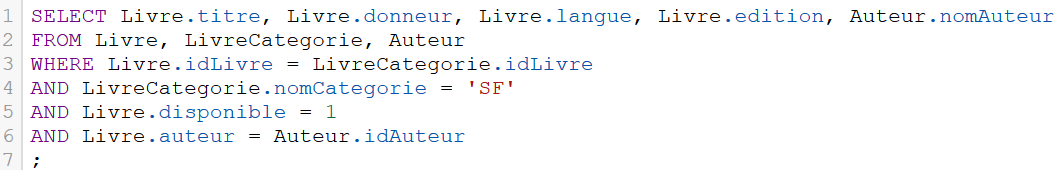


Pour la table profil :



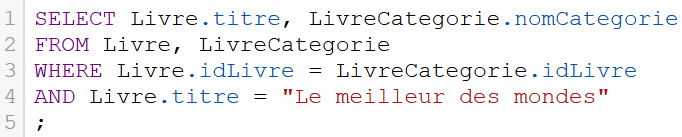
Pour finir voici un exemple de ce que l’on est capable de faire avec notre base de données :

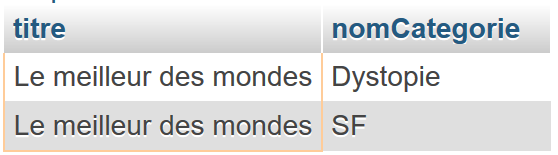
Trouver tous les livres de la catégorie « SF » actuellement disponibles :



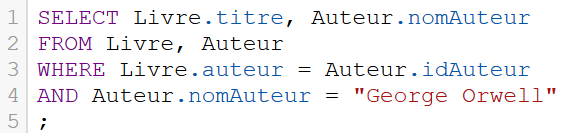


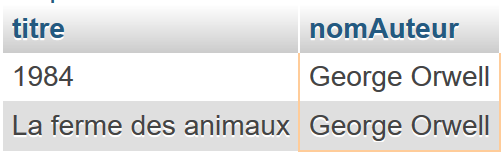
Trouver les categories d’un livre (ex : « Le meilleur des mondes ») :





Trouver tous les livres d’un auteur (exemple George Orwell) :





Trouver les informations des donneurs et des personnes intéressés par leur livre :

