

# GOODREAD

Goodread est un réseau social ayant pour but de partager avec d'autre utilisateur ses goût littéraire, donner son avis sur des œuvres ainsi que découvrir et faire découvrir de nouveaux auteurs.

Etant fréquenté par certain d'entre eux, il est d'ailleur possible d'échanger directement avec des écrivains.

Ce réseau social référence des livres ajoutés par les utilisateurs. Il est ensuite possible à ces dernier d'écrire des publications, les «liker», les commenter ou encore de discuter avec d'autre utilisateurs.

Ce réseau est bien évidemment modéré par des administrateurs qui peuvent sanctionner tout comportement incorrect.

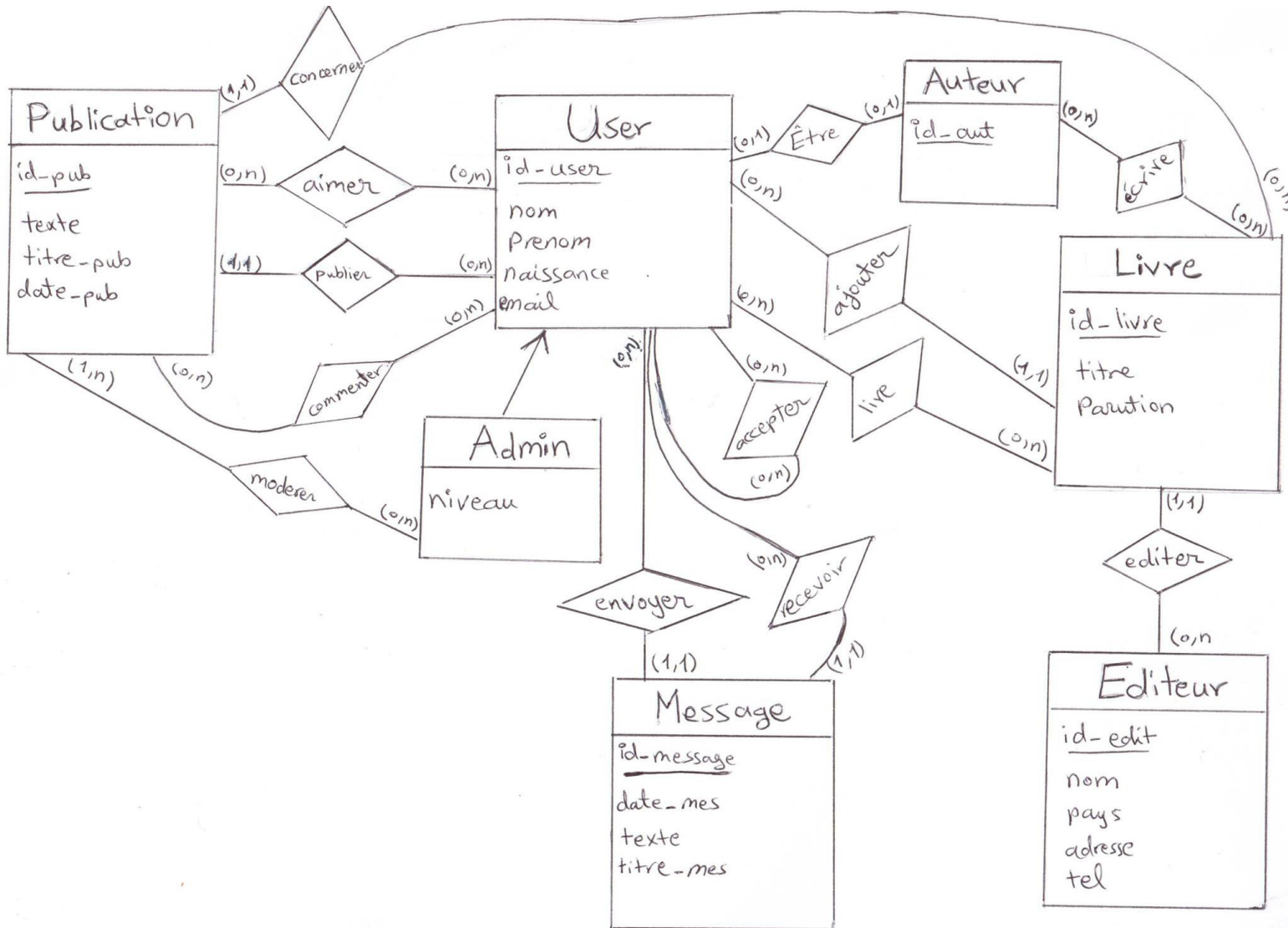
L'utilisateur du réseau social peut éventuellement être écrivain et avoir écrit l'un des livre référencé sur sur le réseau social, d'ou la nécessité de créer une entité Auteur. Cet entité se traduit par une table ayant comme attributs un identifiant d'auteur ainsi qu'un identifiant d'utilisateur représentant ses caractéristiques d'utilisateur.

Un utilisateur a la possibilité de communiquer avec d'autre utilisateur du site. Cette possibilité se traduit par la table «Message» contenant deux clés étrangères vers la table User: «id\_user» représentant l'id de l'expéditeur et «dest» représentant celui du destinataire.

Le fait qu'un administrateur puisse modérer une publication se traduit par la table «Moderee»: celle ci contient l'id des publications ayant effectivement été modérées, ainsi qu'un attribut «modif» contenant des explications sur la nature des changement effectués.

La table Admin comporte un attribut niveau: celui ci différencie les different niveaux d'accès octroyés aux different administrateurs. Deux niveaux sont possible, niveau «1» représentant le niveau le plus bas et le niveau «2».

## Modèle entités-associations



Le premier choix qui a dû être fait concerne l'entité «Admin»: l'administrateur devait-il être considéré comme un utilisateur amélioré ou bien devait-il être représenté par une nouvelle entité?

Ayant décidé que l'administrateur devait avoir les mêmes possibilités que l'utilisateur classique ( poster des publications, des commentaires etc. ) nous avons finalement décidé d'en faire une entité héritant de User, cela nous évitant ainsi de créer des associations redondantes.

L'autre choix a été de choisir une manière de modéliser le fait qu'un utilisateur puisse être écrivain d'un livre: nous avons choisi de représenter l'auteur par une entité à part afin de lui attribuer certaines associations uniques aux auteurs, l'association «écrire» dans ce cas-ci.

# Schéma relationnel

User ( id\_user, nom, prenom, naissance, email)

Admin ( id\_user , niveau)

Livre ( id\_livre , titre , parution, #id\_edit , #id\_user)

Accepte( id\_accepte , id\_user )

Message ( id\_message, date\_mes , texte , titre\_mes , #dest , #id\_user )

Publication ( id\_pub , texte , titre\_pub, date\_pub , #id\_user , #id\_livre )

Aime ( id\_pub , id\_user )

Commentaire ( id\_pub , id\_user , id\_com , texte\_com)

Moderee ( id\_mod , id\_user , id\_pub , modif)

Auteur ( id\_aut , # id\_user )

Ecrit ( id\_livre , id\_aut )

Editeur ( id\_edit , nom , pays , adresse , tel )

Lu ( id\_livre , id\_user )

# Requêtes

**Requête n°1:** Récupère toute les publications d'un utilisateur.

**Requête n°2:** Permet d'obtenir la liste des publications du réseau en commençant par celle qui ont obtenu le plus de «like».

**Requête n°3:** Récupère les publications venant d'utilisateurs dont ils souhaitent recevoir les publications (plus récentes au plus anciennes).

**Requête n°4:** Calcul pour chaque publication la moyenne d'âge des utilisateur qui l'ont appréciée.

**Requête n°5:** Récupère des id d'utilisateur ayant lu les même livre que l'utilisateur passé en paramètre.

**Requête n°6:** Supprime un utilisateur de la base de donnée.

**Requête n°7:** Récupère le nombre de livres lu par un utilisateur donné.

**Requête n°8:** Permet d'obtenir les titres de tout les livres lu par un utilisateur donné.

**Requête n°9:** Récupère l'id de l'utilisateur ayant le plus de livre sur le réseau social.

**Requête n°10:** Obtient le nombre de livre ajouter à la base de donnée par chaque utilisateur.

**Requête n°11:** Permet d'obtenir pour chaque publication le nombre de «like» reçu.

**Requête n°12:** Récupère l'email de l'administrateur ayant modéré une publication passée en paramètre. Si la publication a été modérée par plusieurs administrateur, tous les emails sont récupérés.

**Requête n°13:** Récupère le nom et le nombre de favoris de chaque utilisateur