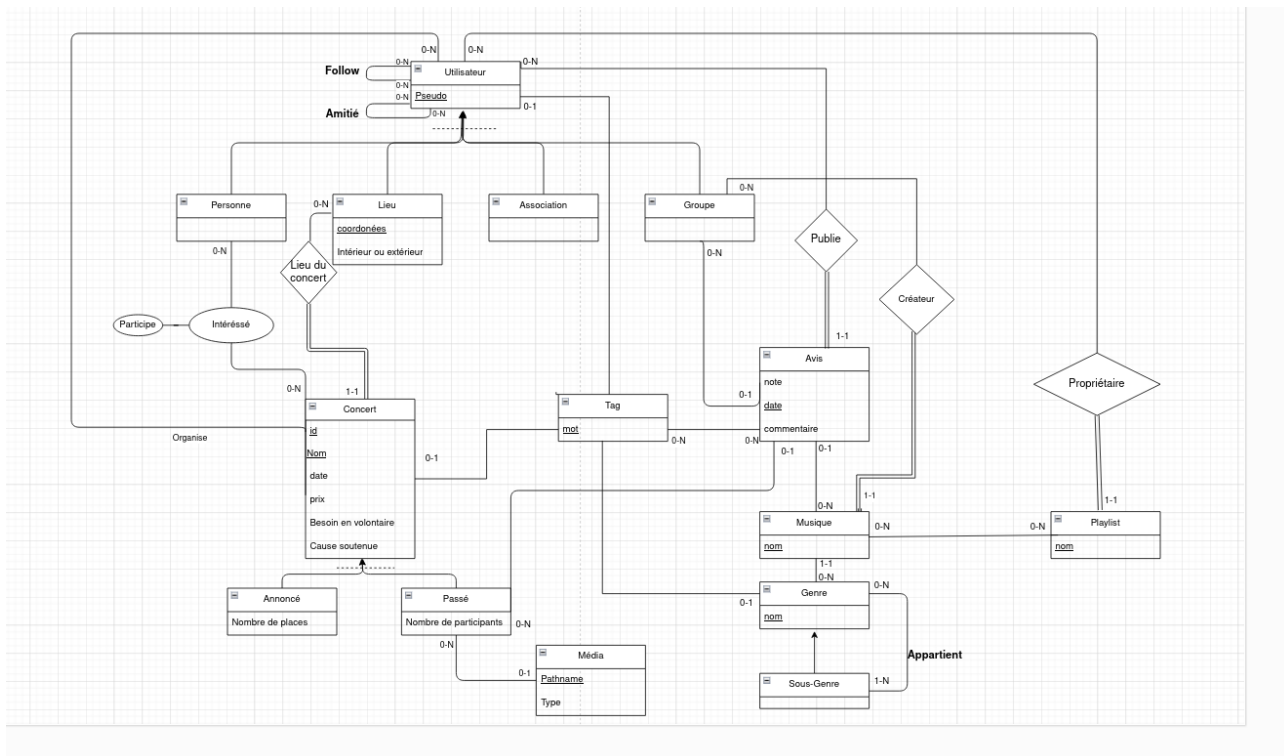


NoiseBook : un réseau social centré sur la musique

Voici le modèle entités-associations que nous avons mis en place pour la représentation de notre base de données.



L'entité concert et ses entités ont été implémentées en une seule table en utilisant l'attribut Nombre de participants pour déterminer si le concert est seulement annoncé ou passé. Si l'attribut était null le concert était seulement annoncé.

L'entité tag s'est vu ajouté un attribut `t_id`, présent comme clé étrangère dans les tables utilisateurs, concerts, playlist, musiques, et genre. Ceci a été fait pour que les noms de ces entités soit équivalent au mot du tag. Utiliser un seul tag par instance de chaque entité était plus pratique à utiliser. Cependant les avis ont tout de même une relation many-to-many avec les tags, on peut donc relier les utilisateurs postant les avis, ou les concerts sur lesquels sont postés les avis, à plusieurs tags relié à l'avis en question.

L'entité utilisateur et ses entités filles ont été implémentées dans des tables différentes sauf pour association car elle ne possède ni relation ni attribut spécifique. La contrainte totale est exprimée par une 'enum' dans la table utilisateur, et à chaque insertion dans utilisateur nous avons un trigger qui check l'enum et insère l'utilisateur automatiquement dans la table fille correspondante.

Les amitiés sont gérées par une paire d'id d'utilisateur toujours ordonnée pour qu'il n'y ai qu'une paire unique et qu'elle soit symétriquement équivalente à l'insertion.

Requêtes implémentées :
(Cf prepare.sql)