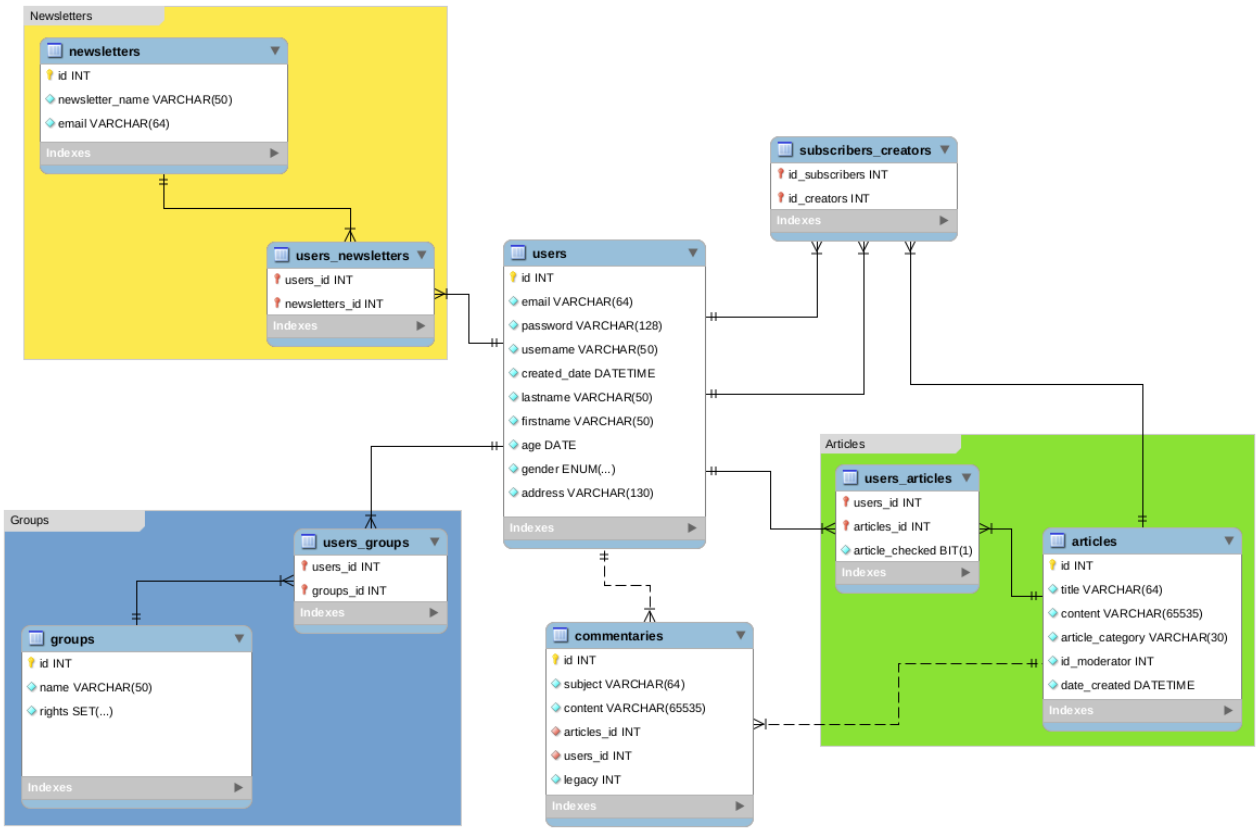
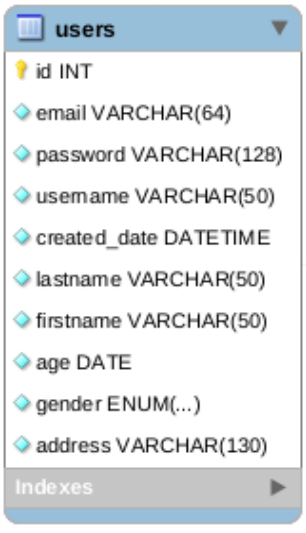
DATABASE MODEL

19MINUTES

avenia_r & dogota_n





Users:

Dans le cas du “password” la taille du VARCHAR dépend du hachage utilisé, dans notre cas nous avons choisi le SHA-512 car c’est le hachage qui retourne la plus longue chaîne de caractères.

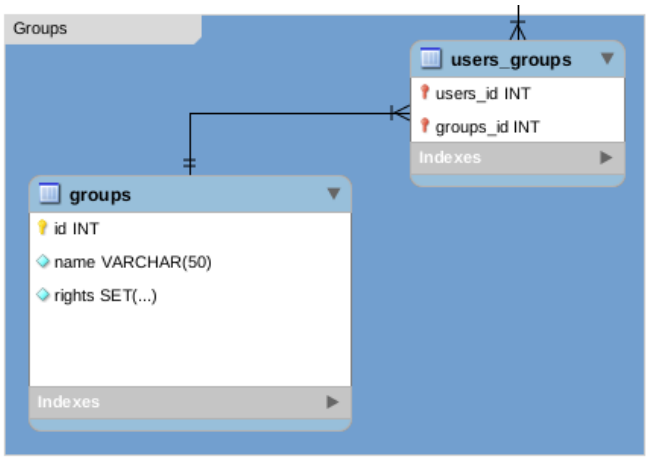
L’âge est calculé à partir de la date de naissance car cela nous permet de calculer l’âge précisément ce qui justifie l’utilisation d’un type DATE.

Nous avons fait le choix d’utiliser un type ENUM pour “gender” car ce type a la particularité d’être utilisé lorsque qu’il n’y a qu’un seul choix possible (contrairement au type SET).

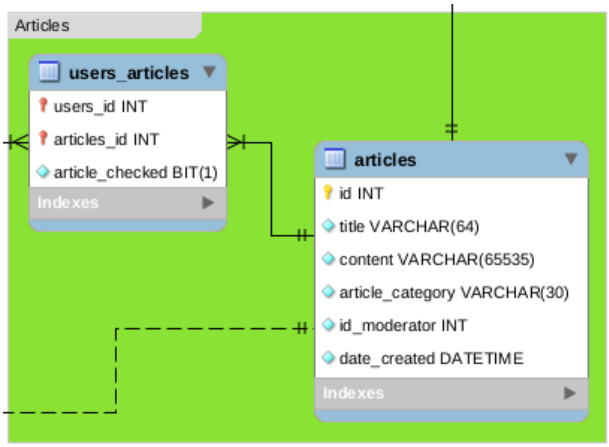
Toutes nos colonnes de la table “users” sont spécifiées en “not null” car les renseignements demandé lors de la création du compte à l’utilisateur existent et possèdent obligatoirement une valeur.

Groups:

Pour “rights” nous avons choisi l’utilisation de SET car ce dernier est utilisé lorsque plusieurs des choix qu’il comprend peuvent avoir une valeur, en l'occurrence ici l’utilisateur à différents types de droits qui lui sont accordés ou non.



La liaison entre les tables “users” et “groups” est de type “many to many” car l’utilisateur peut appartenir à plusieurs groupes et un groupe comprend plusieurs utilisateurs.



Articles:

Nous avons adopté un type VARCHAR avec la particularité d’utiliser la longueur maximale (65535 octets) possible de ce dernier, car cela nous permet de pouvoir contenir la totalité d’un article. Il est préférable d’utiliser le type VARCHAR à TEXT car celui-ci garantit la portabilité de la base de donnée lors du changement du système de gestion de bases de données (SGBD).

Pour “date_created” le type DATETIME est le plus approprié, cela permet de savoir avec précision quand est créé l'article. Et peut permettre par la suite d’intégrer facilement une fonction de trie (par exemple pour un affichage en fonction de l’ancienneté de l’article).

La liaison entre les tables “users” et “articles” est de type “many to many” afin de pouvoir vérifier qu’un utilisateur a déjà vue ou pas un article. Il est nécessaire de passer par une table “users_articles” qui fait la liaison entre les deux et qui permet de connaître si chaque utilisateur a vue ou pas chaque article qui est stocké dans la colonne “article_checked”. Cette dernière est un type BIT(1) qui peut prendre deux valeurs (seulement) différentes.

La table “subscribers_creators” permet de savoir si un utilisateur est abonné à un autre utilisateur, mais aussi de savoir si un l’utilisateur publie un article et donc permet la remontée d’informations vers l’utilisateur abonné à ce dernier. L’utilisation des liaisons “many to many” vers les tables “users” et “articles” est justifié par le fait qu’un utilisateur peut publier plusieurs articles et qu’un utilisateur peut être abonné à d’autres utilisateurs.

Newsletters:

La table “newsletters” s’occupe de savoir à quelle(s) newsletter(s) sont abonnés chaques utilisateurs, d'où la liaison “many to many”. La table “newsletters” possède une colonne “email” qui est nécessaire dans le cas où l’utilisateur s’inscrit à une newsletters sans utiliser de compte, dans le cas contraire on utilisera l’adresse mail du compte qui est déjà renseignée.