

Base de données

La base de données a été construite autour de la table “utilisateurs”. Nous allons détailler la construction des différentes tables de cette base de données, ainsi que les éventuelles relations entre elles, puis nous verrons la normalisation de ces relations.

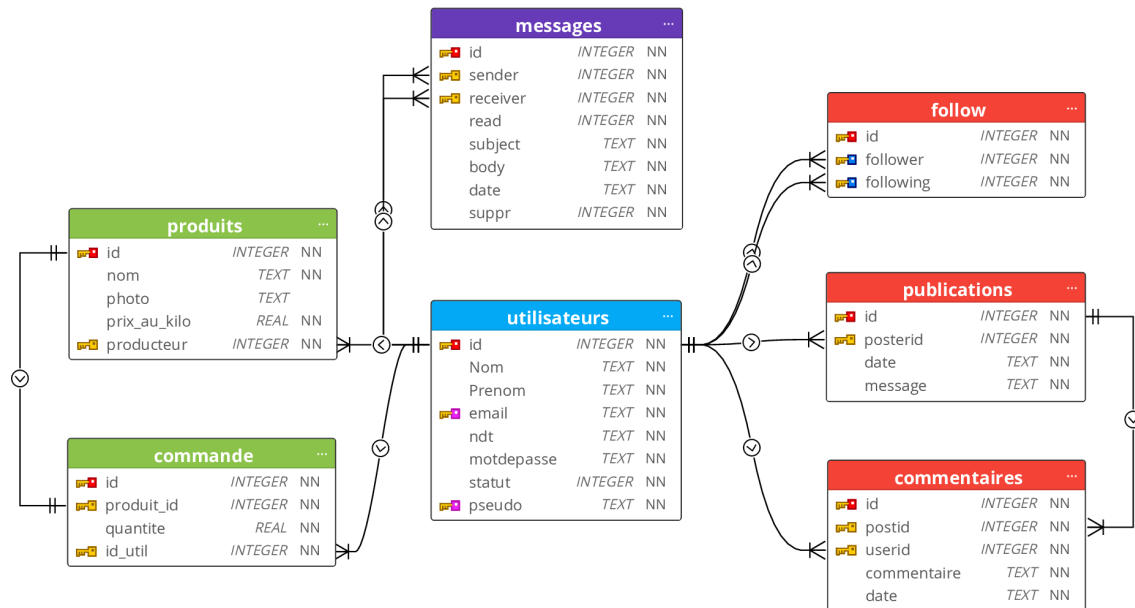


FIGURE 1 – Diagramme de la base de données

1 Construction de la base de données

Dans cette partie, nous allons expliquer la construction de la base de données : signification des différents éléments des tables, et relations éventuelles entre ces tables. La couleur du titre de chaque sous-partie est associée à la couleur du groupe de tables (fig. 1) concerné par cette sous-partie.

1.1 Utilisateurs

Pour chaque inscription, nous demandons un nom, un prénom, un pseudonyme, une adresse e-mail, un numéro de téléphone, un mot de passe, et nous donnons le choix entre un compte producteur ou acheteur. Ces informations étant susceptibles de changer (changement ou erreur dans le nom par exemple), nous avons choisi d'identifier les comptes utilisateurs par un identifiant unique (un entier naturel) attribué automatiquement à l'inscription, et constituant donc notre clé primaire. Toutes ces données sont stockées dans la table “utilisateurs”. De cette manière, toutes les autres tables qui ont un rapport direct avec les utilisateurs peuvent les identifier tout en étant à l'abri d'éventuels changements dans les données des utilisateurs.

1.2 Vente de produits

Les produits qui sont à vendre ou qui ont été vendus par les producteurs sont stockés dans la table “produits”. Un identifiant unique (entier naturel) est affecté à chaque produit, ainsi qu'un nom, un lien vers une photo, son prix au kilo (0 dans le cas d'une vente à l'unité) et l'identifiant du producteur qui vend ce produit.

Les commandes sont stockées dans la table “commande”. Un identifiant unique (entier naturel) est affecté à chacune de ces commandes, qui comportent également l'identifiant du produit acheté, sa quantité (un nombre entier si la vente se fait à l'unité, une masse si la vente se fait au kilo), et l'identifiant de l'utilisateur qui l'a acheté.

1.3 Communication

1.3.1 Fil d'actualité

Les publications des utilisateurs sont stockées dans la table "publications". A chaque publication est affecté un identifiant unique (entier naturel), ainsi que l'identifiant de l'utilisateur qui publie, la date de publication, et le message publié.

Chaque commentaire est stocké dans la table "commentaires", qui comporte un identifiant unique (entier naturel) affecté au commentaire, l'identifiant de la publication commentée, l'identifiant de l'utilisateur qui a commenté, le texte du commentaire et la date d'envoi de ce commentaire.

L'utilisateur choisit quel producteur il "suit", c'est-à-dire de quels producteurs il veut avoir les publications affichées dans son fil d'actualité. Pour cela, la table "follow" recense, pour chaque abonnement, un identifiant unique de cette relation (entier naturel), l'identifiant de l'utilisateur qui suit et celui de l'utilisateur suivi.

1.3.2 Messagerie

Les messages sont stockés dans la table "messages". Chacun de ces messages a un identifiant unique (entier naturel), l'identifiant de l'utilisateur qui envoie le message, l'identifiant de celui qui le reçoit, un entier servant de booléen (car d'après mes recherches, il n'y a pas de type booléen à proprement parler dans sqlite, il faut donc utiliser un entier naturel) qui sert à savoir si l'utilisateur a lu le message ou non (par défaut il est donc sur 0, soit l'état "non lu"), le sujet et le corps du message, la date d'envoi et un entier naturel servant de booléen qui est sur 1 si le receveur du message a décidé de masquer le message (lorsqu'il l'a lu par exemple). Il est donc sur 0 par défaut.

2 Formes normales

Notre base de données respecte les conditions requises pour être en 3NF. Nous allons détailler dans cette partie les différentes raisons pour lesquelles toutes les conditions des 1NF, 2NF et 3NF sont remplies.

2.1 Première forme normale (1NF)

Pour être en première forme normale, une relation doit posséder au moins une clé primaire et tous ses attributs doivent être atomiques.

Comme nous pouvons le voir sur le diagramme (fig. 1), chaque table contient un attribut "id" qui constitue la clé primaire de cette table. Ces attributs "id" sont en auto-incrément. De plus, chaque attribut est atomique : c'est pour cela que la table "commande" a été créée (pour ne pas directement inscrire la liste des commandes dans la table "produits"), ainsi que la table "commentaires" (pour ne pas avoir une liste de commentaires pour chaque publication dans la table "publications"). Il en est de même pour la table "follow", qui sert à ne pas stocker une liste d'utilisateurs suivis pour chaque client dans la table "utilisateurs".

Ainsi, les conditions sont remplies, la base de données respecte la première forme normale.

2.2 Deuxième forme normale (2NF)

Pour être en deuxième forme normale, une relation doit être en première forme normale et tout attribut n'appartenant pas à la clé ne doit pas dépendre d'une partie seulement de la clé. Toutes les relations sont en 1NF d'après la sous-partie précédente. Les clés primaires de toutes les tables sont composées d'un attribut uniquement, donc le respect de la deuxième forme normale est assuré.

2.3 Troisième forme normale (3NF)

Pour être en troisième forme normale, une relation doit être en deuxième forme normale et tout attribut n'appartenant pas à la clé ne doit pas dépendre fonctionnellement d'un attribut qui n'appartient pas à la clé. Toutes les relations sont en 2NF d'après la sous-partie précédente. D'après le diagramme de la base de données et l'explication de la construction de celle-ci dans la partie précédente, dans une table donnée, aucun attribut ne dépend d'un autre, à part de la clé primaire. Le respect de la troisième forme normale est donc assuré.