

Rapport sur la structure de la base de données "social"

Introduction

Dans le cadre du cours de Systèmes de Bases de Données, ce rapport a pour objectif d'analyser la structure de la base de données **social** et de présenter son modèle entité-association (EA) ainsi que son modèle relationnel.

La base de données **social** est conçue pour gérer un système de réseau social avec des utilisateurs, des profils, des publications (Posts), profession, des interactions entre utilisateurs, des centres d'intérêt, etc. L'objectif de ce rapport est de comprendre la conception de la base de données, d'évaluer sa structure et d'identifier les éventuelles limitations.

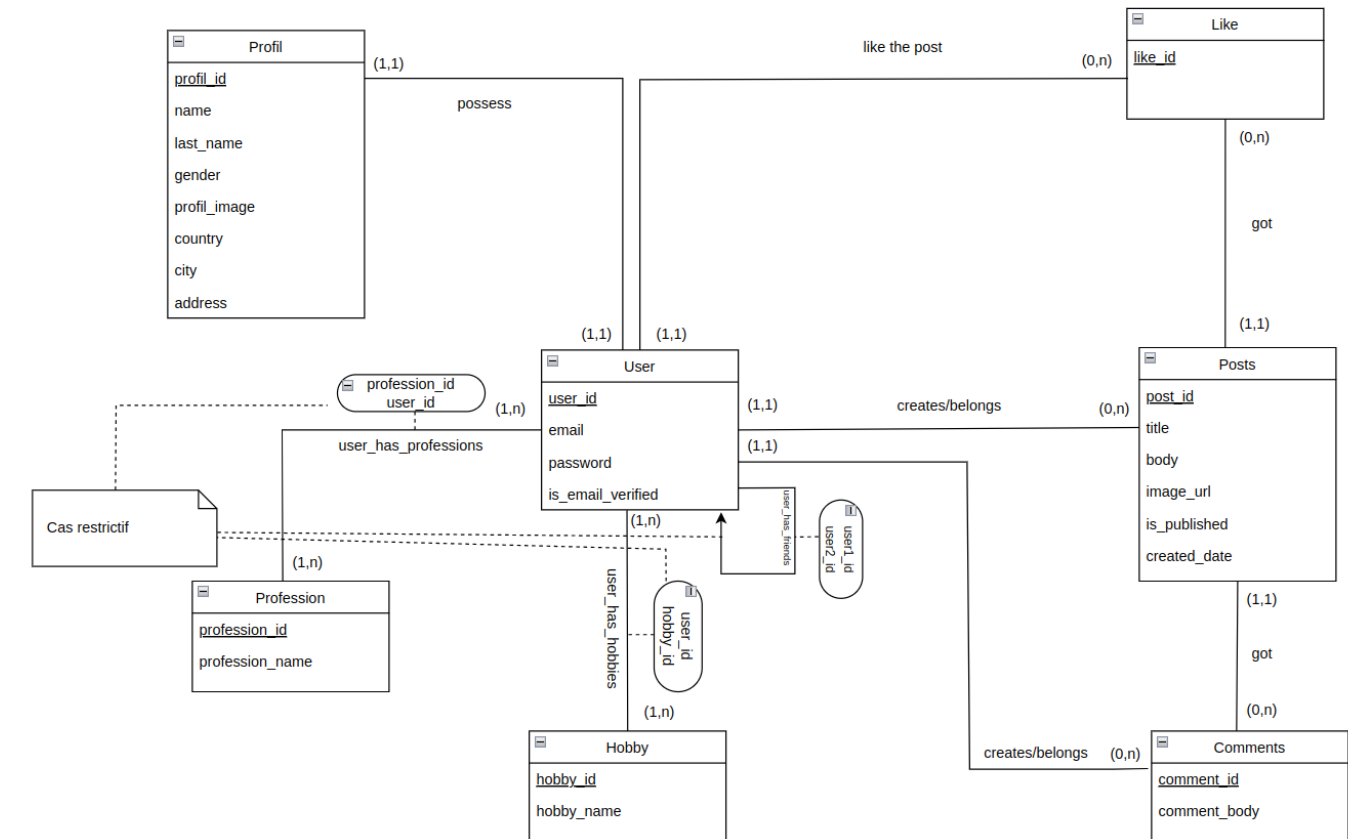
De plus, nous aborderons des aspects tels que les dépendances d'inclusion et d'intégrité, l'indexation et les contraintes non modélisables telles que les déclencheurs. Une analyse complète de ces éléments permettra de mieux comprendre la base de données "social" dans le contexte du cours de Systèmes de Bases de Données.

Modèle entité-association

Le modèle entité-association représente les entités, les attributs et les relations entre les entités. Dans le cas de la base de données **social**, les entités identifiées sont :

- User
- Profil
- Posts
- Comments
- Like
- Hobby
- Profession
- user_has_profession
- user_has_friends
- user_has_hobbies

Le modèle EA permet de visualiser les relations entre ces entités, telles que la liaison entre User et Profil, User et Posts, User et Hobby, etc.



Modèle relationnel

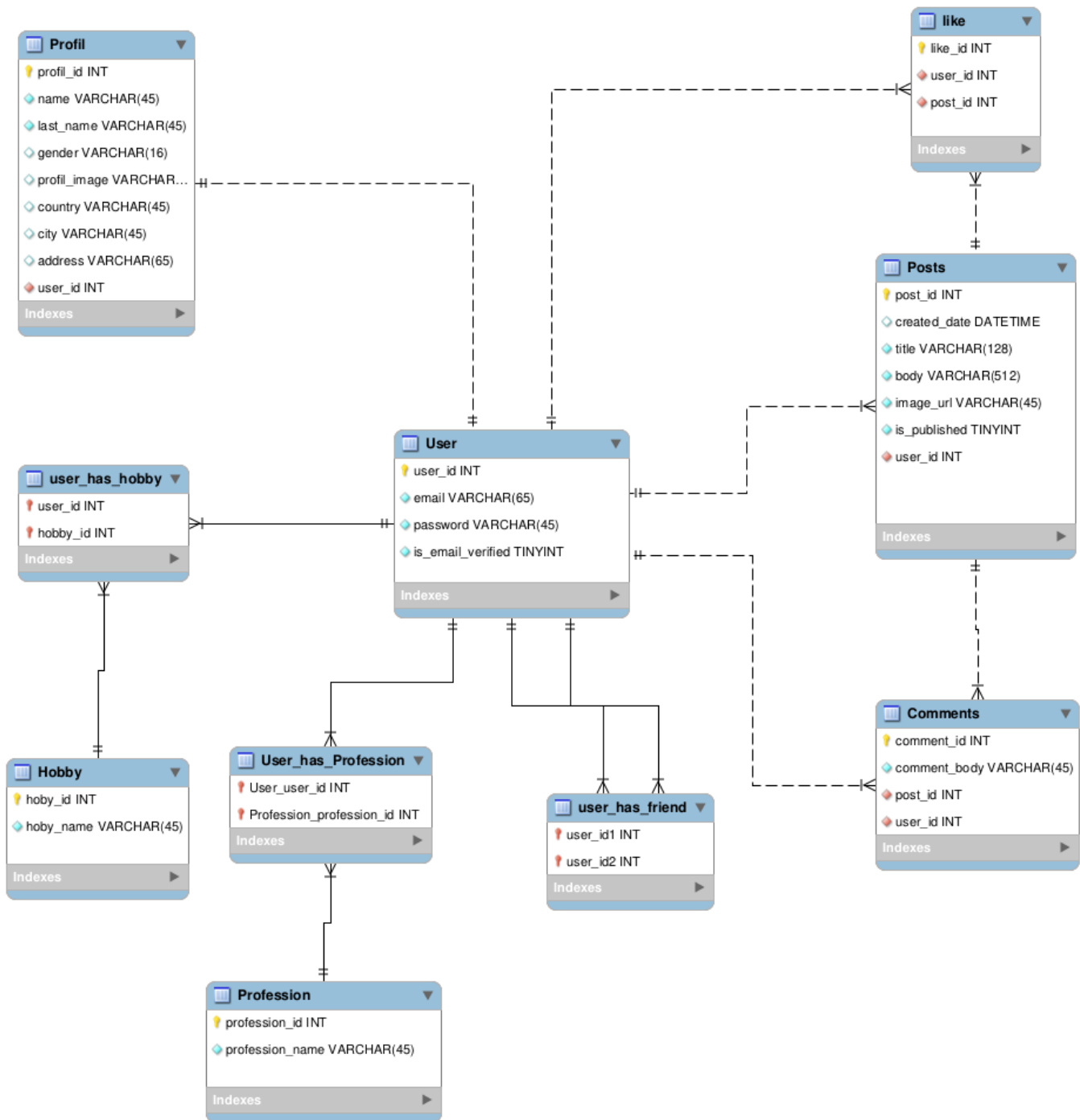
Le modèle relationnel est basé sur les tables suivantes :

- User
- Profil
- Posts
- Comments
- Like
- Hobby
- Profession
- user_has_profession
- user_has_friends
- user_has_hobbies

Les clés primaires et les clés étrangères sont utilisées pour maintenir l'intégrité des données. Cependant, des limitations sont observées, telles que la possibilité de valeurs NULL dans certaines colonnes comme par exemple la photo de profil d'un utilisateur qui peut avoir une valeur **NULL**.

Lorsqu'une colonne autorise les valeurs NULL, cela signifie qu'elle peut contenir des données manquantes ou non définies. Cela peut avoir un impact sur l'application au niveau visuel, notamment si cette colonne est utilisée pour afficher des informations graphiques telles que des images de profil.

Deux solutions possibles pour gérer cette situation sont l'utilisation de déclencheurs (triggers) au niveau de la base de données pour charger une image par défaut lorsque la colonne de la photo de profil est NULL, et la mise en place d'une solution logicielle où une image générique est affichée si la colonne de la photo de profil est NULL. Ces solutions garantissent qu'une image appropriée est toujours affichée pour les profils utilisateur, même en l'absence d'une photo de profil spécifique.



Dépendances d'inclusion et d'intégrité:

Les dépendances d'inclusion et d'intégrité doivent être respectées pour préserver l'intégrité des données. Les dépendances d'inclusion garantissent que les valeurs des attributs respectent certaines conditions. Les dépendances d'intégrité assurent que les relations entre les entités sont maintenues correctement. Dans mon cas, les clés étrangères sont utilisées pour représenter les dépendances d'intégrité entre les tables.

Nécessité d'indexation

L'indexation peut être nécessaire pour optimiser les performances des requêtes. Deux index sont créés, tels que l'index unique pour l'attribut **email** dans la table User et l'index unique pour l'attribut **hobby_name** dans la table Hobby, **profession_name** dans la table profession. Cependant, l'ajout d'index supplémentaires pourrait améliorer davantage les performances, en particulier pour les colonnes utilisées fréquemment dans les requêtes.

Contraintes non modélisables (triggers)

Un déclencheur (trigger) est utilisé pour mettre à jour la colonne **created_date,is_published** dans la table Posts. Ce déclencheur modifie la valeur de la colonne en fonction de la date d'insertion. Si la date n'est pas valide, le déclencheur attribue une valeur 0 à l'attribut **is_published** et la date actuelle à l'attribut **created_date**, ce qui correspond à la non-publication d'un post. Cette contrainte permet de maintenir la cohérence des données.

Conclusion

La base de données **social** présente un modèle EA et relationnel qui permet de représenter les entités, les attributs et les relations de manière globale. Cependant, ce modèle est relativement simple et peut être considéré comme une première étape dans le développement d'un système de réseau social plus complexe et précis.

Des limitations existent, notamment en termes de spécification des contraintes d'intégrité et de modélisation des relations many-to-many. Des améliorations pourraient être apportées pour préciser les rôles et les contraintes entre les entités, ainsi que pour optimiser les requêtes et les performances de la base de données.

L'indexation et l'utilisation de contraintes non modélisables, comme les déclencheurs, sont également nécessaires pour garantir l'intégrité des données et pour améliorer les performances.

En conclusion, le modèle de la base de données **social** est un point de départ pour la construction d'un système de réseau social plus élaboré. Des étapes supplémentaires et une analyse plus approfondie seront nécessaires pour développer un modèle plus précis et adapté aux besoins spécifiques du système social envisagé.