

Conception et implantation d'une Base de Données sous Postgresql

Création de schéma

Imaginons une communauté d'utilisateurs avides de pouvoir communiquer. Il s'agit de donner à chaque utilisateur de la communauté la possibilité de créer des billets (post), et de les publier. Chaque billet est constitué d'un contenu (content) qui est un paragraphe dont la taille est limitée à 128 caractères. Le billet est estampillé (daté) à l'instant même ou il est créé. Il est identifié par un numéro (entier) qui n'a pas de signification particulière qui sera donc produit automatiquement. Chaque billet ainsi produit pourra être commenté par les usagers vers lesquels le billet est publié. la création d'un commentaire rendra le billet ainsi que ses commentaires visibles par tous les amis du commentateur.

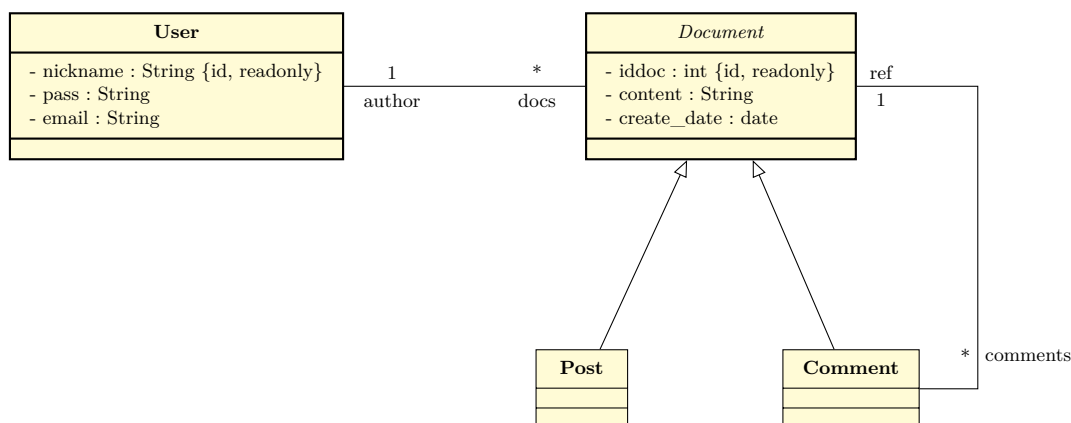


FIGURE 1 – Diagramme de classes

Questions

1. Traduisez le diagramme de classes proposé en un schéma relationnel décrit sous forme de graphe des contraintes d'intégrité référentielles (un sommet, une relation, et un arc, une clé étrangère). Est-ce que vous voyez d'autres contraintes qui ne seraient pas exprimées sur le diagramme de classe ?
2. Implantez ce schéma relationnel sous *PostgreSQL* à l'aide d'un script SQL. Ce script doit permettre de réinitialiser la base à tout moment et automatiquement en créant le schéma et les tables, puis en peuplant la base. Les consignes suivantes devront obligatoirement être respectées :

- Les relations (tables) seront définies au sein d'un schéma "forum1".
- Les noms des relations de base seront définis en minuscules et préfixées par un "_". Exemple : `_user`, `_document`.
- Les contraintes seront nommées explicitement et respecteront les règles énoncées l'an passé.
- Les noms d'attributs seront définis en minuscules et reproduiront les variables d'instance ou noms de rôle (au singulier) du diagramme de classes.

Remarque : les valeurs de "pass", "email", "nickname" seront choisies par l'utilisateur. Les valeurs de dates seront affectées automatiquement par le SGBD.

Initialisation de la base *PostgreSQL*

La base de données sera initialisée manuellement (les commandes INSERT sont à ajouter au script précédent) à l'aide d'un jeu d'essai reproduisant l'exemple suivant. Les valeurs de "pass", "email" sont laissées à votre appréciation. En tête, on trouve l'idoc, le nickname, puis le contenu du document qu'il publie.

Jeu d'essai :

- 1 - Félix - Bonjour tout le monde !
 - |_ 3 - Arthur - Salut Félix ! Tu vas bien ?
 - | |_ 4 - Félix - Oui, merci. Et toi ?
 - |_ 5 - Jean - Tiens, Félix, tu es rentré de vacances ?
 - |_ 6 - Thomas - Bonjour Félix ! Bonjour à tous !
 - |_ 7 - Jean - Salut Thomas !
 - |_ 8 - Arthur - Hello Thomas. Tu reprends le foot cette année ?
 - |_ 11 - Thomas - Non, je vais me mettre au handball, je crois.
 - |_ 12 - Félix - Ah ? Dommage, on s'amusait bien ensemble.
- 2 - Arthur - Quelqu'un jouera encore au foot cette année ?
 - |_ 9 - Jean - Oui, moi !
 - |_ 10 - Thomas - Pas moi.
 - |_ 13 - Félix - Moi aussi.
 - |_ 14 - Alex - Et moi !

Expression de requêtes

Formulez et testez les requêtes suivantes en langage SQL :

1. Listez les utilisateurs.
2. Listez les documents publiés par Félix en les classant par ordre chronologique croissant.
3. Listez la hiérarchie des commentaires associés au post numéro 1 (celui de Félix disant bonjour à tout le monde. S'il possède un identifiant différent, vous adapterez la requête). Cette hiérarchie est composée de l'id d'un commentaire et de l'id du document auquel il répond.
4. Définissez la fonction "commentaires" qui donne la hiérarchie évoquée à la question précédente à partir d'un numéro de document donné en paramètre. Si le document n'existe pas, la fonction retourne un résultat vide et affiche un message d'erreur : "Ce document n'existe pas.". Si le document est une feuille dans l'arbre des documents, la réponse de la fonction est vide.

Mises à jour portant sur la base de données

- Une opération de suppression d'un utilisateur n'entraînera pas la suppression des documents correspondants. La solution consiste à affecter les documents de l'utilisateur supprimé à un utilisateur virtuel nommé "Anonymous". Ainsi, les documents associés à cet utilisateur resteront visibles et seul le nickname affiché changera.