

# Modélisation d'une base de données d'un blog

## Groupe :

- ❖ HEGO Nathan
- ❖ MICHON Alexandre
- ❖ JOURAND Benjamin
- ❖ BANCE Florent (absent ce jour)

## Présentation de la base de données :

Ce modèle de BDD servira pour l'élaboration d'un blog.

- ❖ La base comportera 5 tables :
  - T\_USERS : table comprenant les utilisateurs, leur(s) rôle(s) et leur(s) droit(s).
  - T\_ARTICLES : table comprenant les articles.
  - T\_COMMENTAIRES : table comprenant les commentaires.
  - T\_CATEGORIES : table comprenant les catégories des articles.
  - T\_MEDIAS : table comprenant les médias contenus dans les articles.

## Présentation de la table T\_USERS :

La table T\_USERS comprend les champs :

- USER\_ID de type INT : celui-ci sert d'index auto-incrémenté permettant d'identifier l'utilisateur (clé primaire). Le choix de déclarer le champ en INT découle du fait qu'il pourrait y avoir une multitude d'utilisateurs du blog.
- 
- USER\_PSEUDO de type VARCHAR(50) (Chaîne de 50 caractères maximum) : désigne le pseudo de l'utilisateur (Champs obligatoire car déclaré comme NON-NULL et unique avec l'option UNIQUE)
- USER\_MAIL de type VARCHAR(250) (Chaîne de 250 caractères maximum) : désigne le mail de l'utilisateur (Champs obligatoire car déclaré comme NON-NULL et unique avec l'option UNIQUE).
- USER\_MOT\_DE\_PASSE de type CHAR(50) (Chaîne de 50 caractères maximum) : désigne le mot de passe choisi par l'utilisateur (Champs obligatoire car déclaré comme NON-NULL). Ce champ est de format CHAR pour pouvoir le hashé en SHA1.

- USER\_NOM de type VARCHAR(50) (Chaine de 50 caractères maximum) désigne le nom de l'utilisateur (Champs obligatoire car déclaré comme NON-NULL).
- USER\_PRENOM de type VARCHAR(50) (Chaine de 50 caractères maximum) désigne le prénom de l'utilisateur (Champs obligatoire car déclaré comme NON-NULL).
- USER\_AVATAR de type VARCHAR(255) désigne l'URL du serveur de stockage du fichier image servant pour l'avatar de l'utilisateur.
- USER\_ROLE de type INT() (entier) désigne le rôle de l'utilisateur (Champs obligatoire car déclaré comme NON-NULL). Nous pouvons imaginer que chaque rôle serait défini par un entier court et être utilisé une fois par utilisateur (niveau d'autorisation). Par exemple, le 0 pour les administrateurs car il peut y avoir plusieurs administrateurs sur le blog (droit de poster des articles, des modérer des commentaires et gérer les droits des autres utilisateurs), 1 pour les utilisateurs abonnés (possibilité de lire l'article et de soumettre un commentaire à valider par les administrateurs).

Dans le futur, nous pourrions imaginer des niveaux d'utilisateurs avec des contenus gratuits et des contenus payant et dans ce cas, ajouter une nouvelle valeur de ce champ et de nouveaux droits de lecture)

- USER\_DATE\_CREATION de type DATETIME : désigne la date de création du compte utilisateur (Champs obligatoire car déclaré comme NON-NULL).
- USER\_DATE\_DERNIERE\_CONNEXION de type DATETIME : désigne la date de la dernière connexion de l'utilisateur (préparation pour le RGPD et le stockage de données personnelles -> si pas de connexion depuis 3 mois possibilité de suspendre le compte utilisateur) (Champs obligatoire car déclaré comme NON-NULL).
- USER\_RECEVOIR\_NOTIFICATIION de type TINYINT : désigne le souhait de recevoir une notification si un article est publié sur le blog. Valeur 0 ou 1 suivant si l'utilisateur le souhaite (1) ou non (0).

#### Présentation de la table T\_ARTICLES :

- ARTICLE\_ID de type INT : celui-ci sert d'index auto-incrémenté permettant d'identifier l'article (clé primaire).
- ARTICLE\_TITRE de type VARCHAR (255) : désigne le titre de l'article (Champs obligatoire car déclaré NON-NULL et unique car déclaré avec option UNIQUE).
- ARTICLE\_SOUSTITRE de type VARCHAR(255) : désigne le sous-titre de l'article.
- ARTICLE\_CONTENU de type LONGTEXT : désigne le contenu de l'article. Celui-ci étant de longueur inconnue, nous avons pris parti de mettre la taille maximum alloué au texte.

- ARTICLE\_DATE\_CREATION de type DATETIME : désigne la date de création de l'article (champ obligatoire car déclaré NON-NULL).
- ARTICLE\_DATE\_MODIFICATION de type DATETIME : désigne la date de modification de l'article.
- ARTICLE\_DATE\_PUBLICATION de type DATETIME : désigne la date de publication de l'article.
- ARTICLE\_ETAT de type TINYINT : désigne le statut de publication. Nous pourrions imaginer le statut 0 pour le stade de brouillon, 1 pour le stade publié et pourquoi pas 2 pour le stade archivé (donc possibilité de le retrouver dans une recherche).
- ARTICLE\_MEDIA de type VARCHAR(255) : désigne l'ID du média ( lien n à m à la table T\_MEDIAS) joints à l'article. Un article peut contenir plusieurs médias et un media peut être contenu dans plusieurs articles différents.

#### Présentation de la table T\_COMMENTAIRES :

- COMMENTAIRE\_ID de type INT : index auto-incrémenté permettant d'identifier le commentaire (clé primaire).
- COMMENTAIRE\_MESSAGE de type LONGTEXT : désigne le message du commentaire.
- COMMENTAIRE\_DATE\_CREATION de type DATETIME désigne la date de création du commentaire.
- COMMENTAIRE\_ETAT\_MODERATION de type TINYINT désigne si le commentaire est approuvé (1) ou non (0) par défaut nous pourrions mettre cette valeur à 0.
- COMMENTAIRE\_MODERATEUR de type INT : désigne l'ID de l'utilisateur modérateur qui a approuvé ou non le commentaire.

#### Présentation de la table T\_CATEGORIES :

- CATEGORIE\_ID de type INT : index auto-incrémenté permettant d'identifier la catégorie (clé primaire).
- CATEGORIE\_NOM de type VARCHAR(255) désigne le nom de la catégorie.

Un article peut avoir plusieurs catégories et une catégorie peut contenir plusieurs articles (relation de n catégories à m articles).

### Présentation de la table T\_MEDIAS :

- CMADIA\_ID de type INT : index auto-incrémenté permettant d'identifier la catégorie (clé primaire).
- MEDIA\_URL de type VARCHAR(255) désigne l'url de la source du media. (Lien n à m à la table T\_ARTICLES). Un article peut contenir plusieurs médias et un media peut être contenu dans plusieurs articles différents.

### EXPLICATION DES LIENS :

1 utilisateur (type administrateur) peut écrire plusieurs articles mais un article n'est lié qu'à un seul utilisateur (lien 1 utilisateur à N articles)

1 utilisateur (type administrateur) peut modérer un ou plusieurs commentaires mais un commentaire ne peut être modéré que par un seul utilisateur (modérateur) et 1 utilisateur peut écrire plusieurs commentaires mais un commentaire ne peut être associé qu'à un seul utilisateur. (Lien 1 utilisateur à N commentaires)

1 commentaire ne peut être associé qu'à un seul et unique article par contre un article peut avoir plusieurs commentaires. (Relation de 1 article à N commentaires)

1 article peut être associé à plusieurs catégories mais 1 catégorie peut être associée à plusieurs articles. (Liaison de N catégories à M articles).

1 article peut contenir plusieurs médias mais 1 média peut être contenu par plusieurs articles (exemple une vidéo de présentation ou un logo...). (Lien N médias à M articles).

Il y a sûrement une grande amélioration à faire sur ce modèle, à suivre...