

TD DONNÉES 1 - Jointures

L1 - HEI Madagascar

(D'autres requêtes peuvent être demandées verbalement lors du TD.)

Sujet 1 : Un simple blog

Pour ce premier exercice, nous allons concevoir un blog où les utilisateurs peuvent publier librement du contenu. Afin de guider la définition des contraintes, voici quelques règles de gestion :

- **RG1** : Une publication est associée à un seul utilisateur. Il ne doit pas être possible d'avoir une publication sans auteur.
- **RG2** : Lorsqu'une publication n'a pas de titre, celui-ci doit être défini par défaut à "Untitled".
- **RG3** : Un utilisateur doit obligatoirement posséder un nom. Son adresse e-mail est facultative, mais si elle est renseignée, elle doit être unique.
- **RG4** : Le genre d'un utilisateur doit être l'une des valeurs suivantes : "M" (Masculin), "F" (Féminin) ou "X" (Autre).

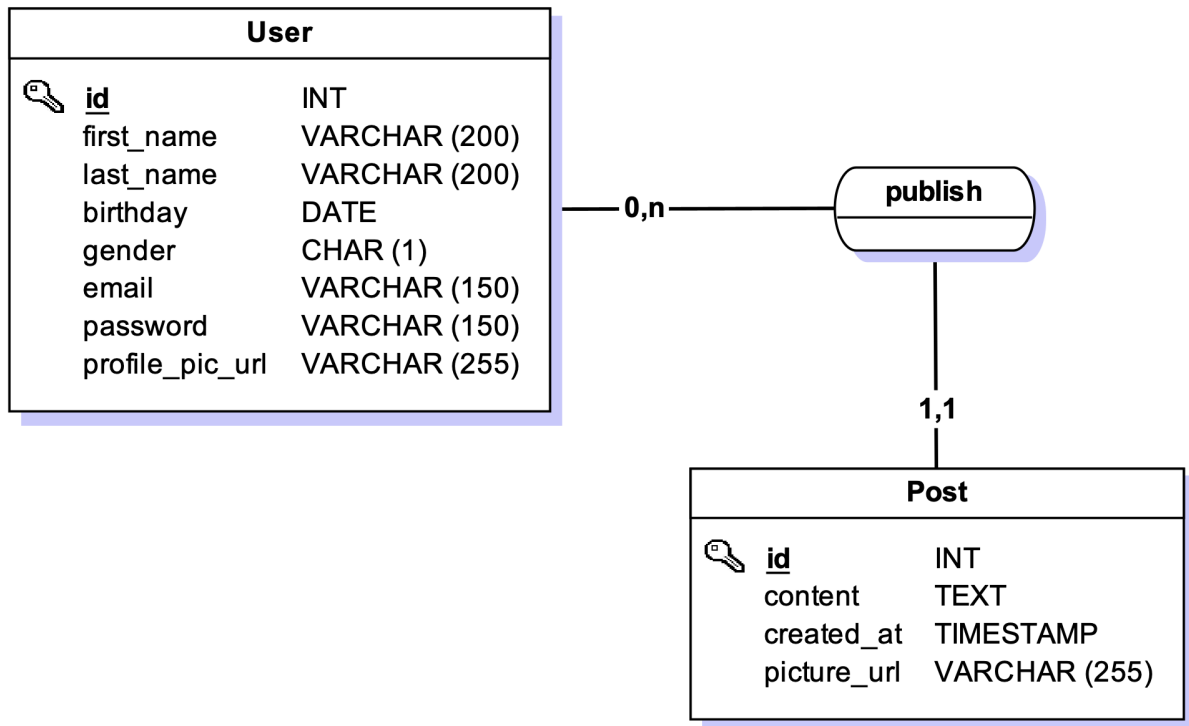


Figure 1 : MCD du blog

Travail demandé

1. Écrire le script SQL permettant d'implémenter cette base de données en respectant les contraintes.
2. Insérer suffisamment de données pour tester les requêtes de la question 3.
3. Écrire les requêtes permettant d'obtenir les informations suivantes :
 - Récupérer tous les utilisateurs dont le nom commence par "Andria", en tenant compte de la casse, triés par identifiant croissant.
 - Récupérer tous les utilisateurs dont le nom, le prénom ou l'adresse e-mail contient "RAK", sans distinction de casse, triés par identifiant croissant.
 - Récupérer tous les utilisateurs dont l'e-mail ou le mot de passe est nul.
 - Récupérer toutes les publications avec le nom et le prénom de leurs auteurs respectifs, triées par date de publication décroissante.
 - Récupérer toutes les publications du jour, avec le nom et le prénom de leurs auteurs respectifs, triées par date de publication décroissante.
 - Récupérer toutes les publications des sept derniers jours, avec le nom et le prénom de leurs auteurs respectifs, triées par date de publication croissante.

- Supprimer toutes les publications des dernières 24 heures appartenant à l'utilisateur dont l'identifiant est 1.

Sujet 2 : Revue de restaurants

Nous allons maintenant spécialiser le blog en un système d'avis sur les endroits pour manger, à l'image de *Tiakaly* et *MACH* (*Mes Aventures Culinaires et Hôtelières* sur Facebook).

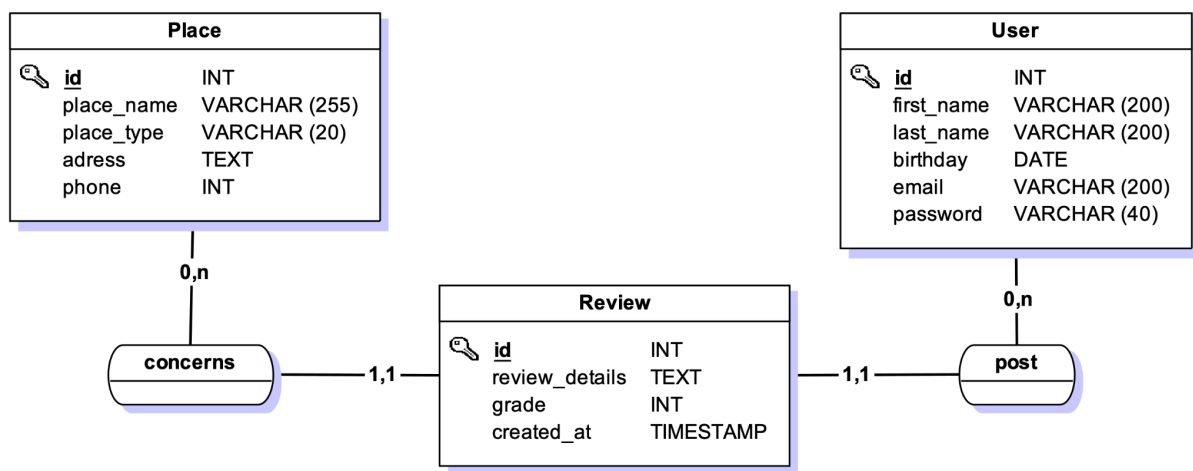


Figure 2 : MCD du système d'évaluation des restaurants

Règles de gestion

- **RG1** : Chaque utilisateur peut attribuer une note de 0 à 5 à un endroit. Les notes peuvent être des nombres décimaux (ex. : 3.2, 4.5).
- **RG2** : Afin d'éviter les avis abusifs, chaque avis doit être associé à un utilisateur, et l'adresse e-mail de chaque utilisateur doit être unique.
- **RG3** : Chaque avis peut être commenté, et chaque commentaire peut lui-même être commenté.
- **RG4** : Un lieu évalué peut être de l'un des trois types suivants : *Restaurant*, *Gargote*, *Bar*.

Travail demandé

1. Écrire le script SQL pour implémenter cette base de données.
 2. Les commentaires et leurs reponses ne sont pas encore sur ce MCD, proposez un moyen pour les modéliser en ajoutant de nouvelles entités et/ou relations au MCD ;
 3. Modifier le type de la colonne **phone** dans la table **utilisateur**, en passant de **INT** à **VARCHAR(15)**.
 4. Écrire les requêtes permettant d'obtenir les informations suivantes :
 - Récupérer les avis (toutes les colonnes de la table **avis**) sur un lieu spécifique (**id = 1**) dont la note est supérieure ou égale à **4**.
 - Récupérer les avis sur un lieu spécifique (**id = 1**) dont la note est inférieure ou égale à **2**, publiés au cours des 30 derniers jours, triés par identifiant croissant.
 - Récupérer les avis sur les restaurants et gargotes contenant le mot "**glacé**", sans distinction de casse. Les résultats doivent être triés par nom croissant des restaurants, suivis des gargotes également triées par nom croissant. Les colonnes affichées seront : nom du lieu, contenu de l'avis, date de publication de l'avis, nom et prénom de l'auteur.
 - Récupérer les avis en affichant le nom du lieu, le contenu de l'avis, la date de publication, et le nom et prénom de l'auteur, triés par ordre alphabétique du nom du restaurant.
 - Récupérer les dix avis les plus récents avec ces mêmes informations, triés par ordre alphabétique du nom du lieu.
-

Sujet 3 : Gestion d'une salle de cinéma

Nous allons concevoir une base de données pour gérer les projections de films en salle de cinéma.

Règles de gestion

- **RG1** : Chaque projection est associée à un film spécifique.

- **RG2** : Chaque projection a lieu dans une salle à une date et une heure précises.
- **RG3** : Chaque salle possède une capacité maximale de spectateurs et un nom unique.
- **RG4** : Un client peut effectuer plusieurs réservations, mais chaque réservation est associée à un seul client.
- **RG5** : Chaque film appartient à un ou plusieurs genres, qui peuvent être définis et modifiés.
- **RG6** : Une réservation peut être en cours ou annulée.

Travail demandé

1. Proposer un MCD adapté.
 2. Écrire les requêtes permettant d'obtenir les informations suivantes :
 - Afficher la liste des films contenant "**héros**" ou "**guerre**" dans leur description.
 - Afficher toutes les réservations sur une projection spécifique qui ne sont pas annulées.
 - Afficher toutes les projections du jour avec le titre du film, le synopsis, le genre, la date et l'heure de projection, triées par date et heure croissantes.
 - Afficher tous les films qui n'ont jamais été projetés.
-

Sujet 4 : Recherche d'appartement

Trouver un appartement à Antananarivo peut être compliqué. Nous allons donc concevoir une base de données pour centraliser ces informations.

Travail demandé

1. Proposer un MCD adapté.
2. Implémenter le MLD correspondant.
3. Proposer au moins trois requêtes intéressantes exploitant cette base de données (le CRUD basique ne compte pas). Cet exercice teste votre créativité dans l'utilisation des données stockées.