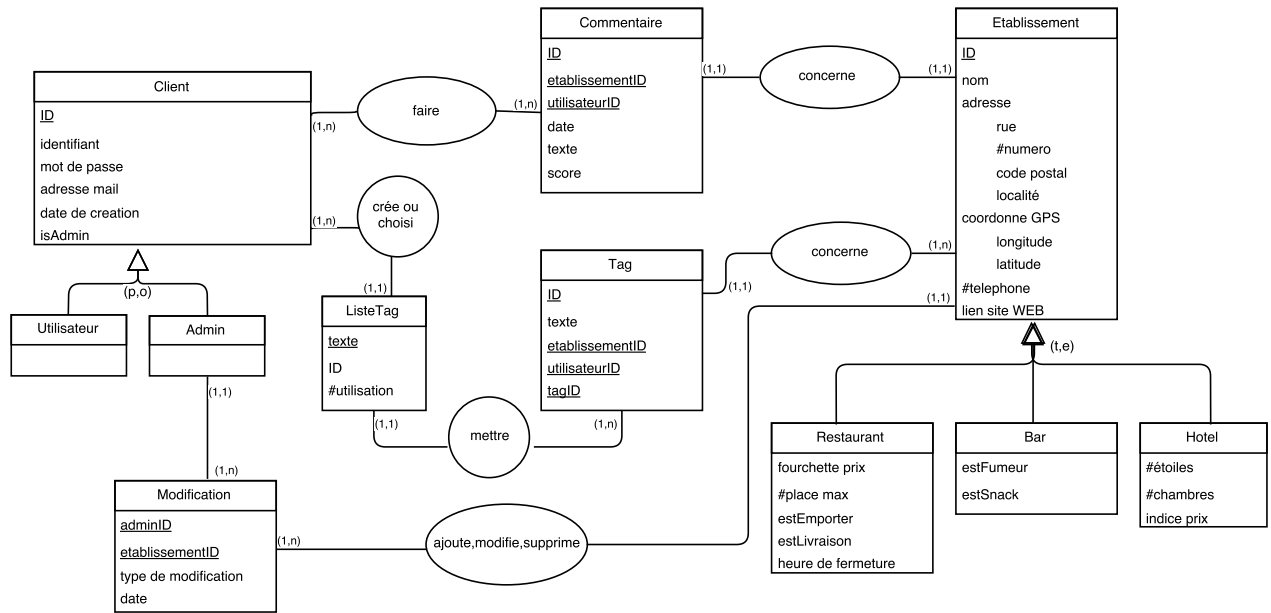


Rapport : Projet - Base de donnée

Maximilien ROMAIN (000411776) - George RUSU (000407965)

10 avril 2016

1 Diagramme



1.1 Contraintes :

- Un client peut être un admin ou bien un utilisateur
- Un client ne peut pas commenter un même établissement une même date
- Un client ne peut pas choisir le même label plusieurs fois pour un même établissement dans un commentaire
- La date d'enregistrement d'un utilisateur doit être inférieure à la date de chaque commentaire
- Un client peut faire un commentaire sans choisir de label(tag)
- Un commentaire possède un ID unique et doit contenir l'ID du client, l'ID du label si il y en a un, l'ID de l'établissement ainsi que le texte et le score.
- Un client ne peut pas labeliser plusieurs fois un même établissement avec un même label

2 Model Relationnel

- Etablissement(ID, Nom, Adresse, AdRue, AdNuméro, AdCodePostal, AdLocalité, Coordonnée, CoodLongitude, CoordLatitude, NumTelephone, *SiteWeb*)
- Restaurant(ID, FourchettePrix, NbrPlace, *Emporter*, *Livraison*, Fermeture)
ID référence Etablissement.ID
- Bar(ID, *Fumer*, *Snack*)
ID référence Etablissement.ID
- Hotel(ID, NbrEtoiles, NbrChambres, IndicePrix)
ID référence Etablissement.ID
- Client(ID, Identifiant, AdresseMail, MotDePasse, DateCréation)
- Utilisateur(ID)
ID référence Client.ID
- Admin(ID)
ID référence Client.ID
- Modification(AdminID, EtablissementID, TypeModif, Date)
AdminID référence Client.ID ; EtablissementID référence Etablissement.ID
- Commentaire(ID, Score, Texte, Date, EtablissementID, UtilisateurID)
EtablissementID référence Etablissement.ID ; UtilisateurID référence Client.ID
- Tag(ID, IDLabel, EtablissementID, UtilisateurID, IDListeTag)
EtablissementID référence Etablissement.ID ; UtilisateurID référence Client.ID ; IDLabel référence ListeTag.ID
- ListTag(ID, countUser, text)
- ClientCommentaire(ID, EtablissementID)
ID référence Client.ID ; EtablissementID référence Commentaire.EtablissementID

3 Choix d'implémentation

Pour expliquer notre schema UML plus haut, nous avons choisi l'implémentation suivante : lorsqu'on voudra connaître les informations d'un établissement en particulière (commentaires, tags , auteur du commentaire, auteur du tag, etc) il nous suffira de retrouver son id correspondant et de regarder avec cet id dans les tables Commentaire et Tag. Idem, pour un utilisateur, si on souhaite savoir ce qu'il a commenté. Nous pouvons donc dire que nos deux tables de bases sont Commentaire et Tag.

En ce qui concerne l'ajout d'un label, nous avons decidé d'avoir une table ("ListeTag") qui va stocker tous les tag/label deja crée avec chacun leur id et chaque label qui caracterisera un etablissement dans la table Tag.

En ce qui concerne le distinction entre utilisateur et admin, nous avons une colonne, isAdmin qui grace a un boolean ou bien a un int (0 ou 1) nous permet de savoir si le client est admin ou bien un simple utilisateur. L'héritage présent dans le schéma permet donc de voir ce qu'un admin peut faire en plus qu'un client standard. C'est même opération supplémentaire que peut faire un admin sont stocké dans une nouvelle table (Modification) qui va permettre de récupérer chaque modification d'un établissement, c'est à dire sont type de modification (Création, suppression ou modification) ainsi que sa date, l'admin qui a opéré et l'établissement concerné.