

Projet BD-IHM : création d'une base de données pour un logiciel de communications

PASSAGE AU SCHEMA LOGIQUE

Nous avons réalisé le schéma logique de notre base de données à partir du schéma conceptuel, en se souciant de la troisième forme normale.

L'association Serveur-Channel est une association 1 : N, un serveur possède plusieurs channels tandis qu'un Channel appartient à un seul serveur.

En suivant les règles de dérivation, nous avons donc ajouter un attribut id serveur dans la classe Channel. Le serveur a un id qui est sa clé primaire, de même pour le Channel. L'attribut id_serveur est une clé étrangère pour la classe Channel, elle référence le Serveur.

La classe Message possède une clé primaire qui est son id, la classe possède plusieurs clés secondaires (id_membre, id_message_ref, id_channel).

Nous avons utilisé une association 0 : N, pour le commentaire, un message peut commenter zéro ou plusieurs tandis qu'un message peut être commenté par zéro ou plusieurs messages. Nous avons transposé cette association dans le schéma conceptuel en insérant l'attribut id_message_ref dans la classe Message, cet argument référence l'id de la classe Message, c'est-à-dire l'id du message qu'il commente.

Dans cette classe nous avons aussi deux autres clés étrangères car le message est lié par deux autres associations de type 0 : N ou bien 1 :N avec respectivement les classes Channel et Message.

Nous avons donc ajouter dans la classe Message, les deux clés étrangères id_channel et id_membre qui référencent respectivement les classes Channel et Membre.

C'est le même principe qui s'applique avec la classe Emoji, qui contient deux clés primaires avec id_membre et id_mess, qui référencent respectivement les classes Membre et Serveur.

La classe Membre est plus complexe : elle comporte deux associations réflexives avec les liens d'amitié et des demandes d'amis.

Il s'agit de deux associations 0 : N O :N , nous avons donc créer deux nouvelles classes, relativement aux règles de dérivations, : les classes Ami et Demande d'Ami.

Ces deux classes comportent deux clés étrangères chacune, qui référencent la classe Membre (Membre(id)). Les classes Demande Ami et Ami ont pour clé différentielle, le couple des deux clés étrangères, qui identifient de manière unique la relation.

Nous avons fait de même avec l'association d'autorisation d'accès à un Channel, où nous avons transformer l'association 2 : N -0 : N en la création d'une nouvelle table Autorisation qui a pour clé primaire, deux clés étrangères qui référencent soit la table Rôle / la table Channel.

En utilisant le même principe, nous avons créé une table intermédiaire entre Rôle et Droit pour modéliser l'association Permission.

FORME NORMALE

Nous avons choisi d'utiliser un seul attribut pour la majorité de nos classes (Serveur, Membre, Channel...) en essayant de respecter, En effet, chaque attribut que nous avons utilisé est atomique. Nos clés primaires sont majoritairement primaires, nous n'avons de problèmes de clés secondaires. Seules les classes Ami et Demande Ami, par exemple sont par exemple d'un couple d'id. Cependant ces classes ne comportent pas d'autres arguments que leurs clés étrangères qui forment ensemble une clé primaire.

Finalement, nous avons fait attention dans nos classes, à respecter la 3^{ème} forme normale, en s'assurant que chaque attribut non clé ne dépende pas d'autres attributs non clés.

Nous avons personnalisé au maximum les différentes classes tout en respectant la 3NF, comme par

exemple avec la classe Message. Elle contient plus de 4 attributs dont le contenu du message, la date, l'heure et le lieu d'envoi. Cependant aucun de ces attributs non compris dans la clé dépend de l'autre.