

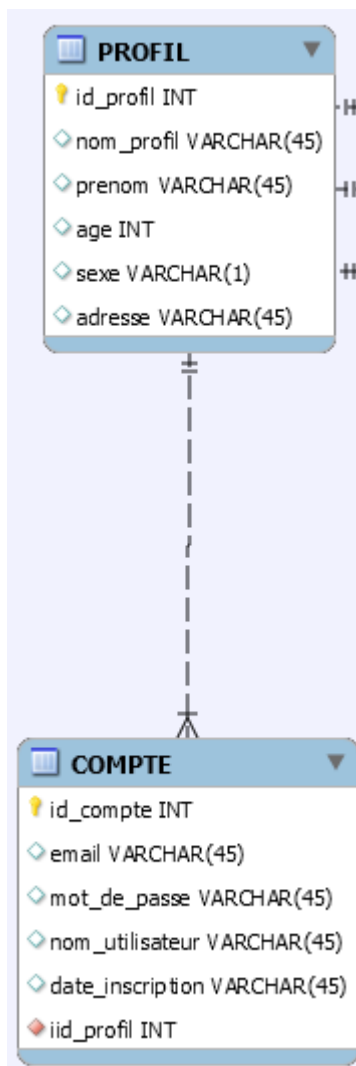
EXPLICATION CAS : 19 MINUTES

Premier Point : Création des comptes UTILISATEURS ainsi que les groupes associés.

Nous avons crée deux tables qui désigne un utilisateur qui viendra sur le site, et cet Utilisateur aura besoin de créer un compte et à son compte sera attaché son profil.

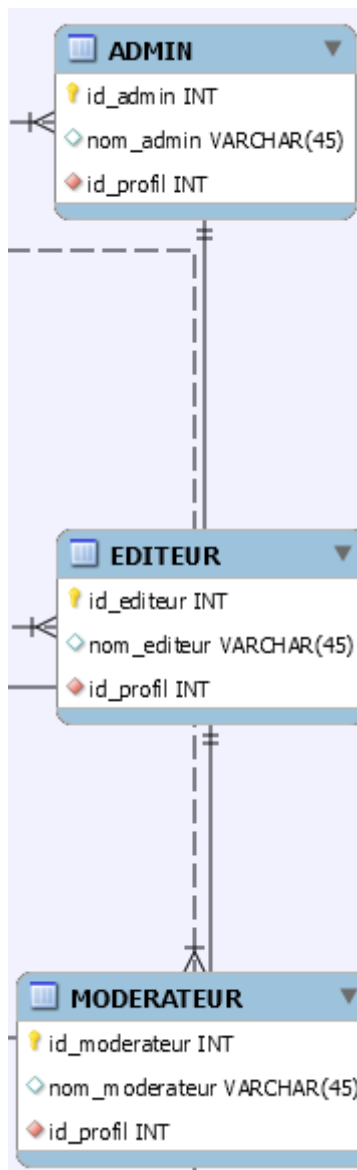
Nous avons définie ici la relation (1 : n) qui nous permet de dire qu'un ou plusieurs profils peuvent avoir un compte, mais un seul compte est attaché à un profil.

CRÉATION DES TABLES POUR UTILISATEURS



CRÉATION DES TABLES POUR LES GROUPEES

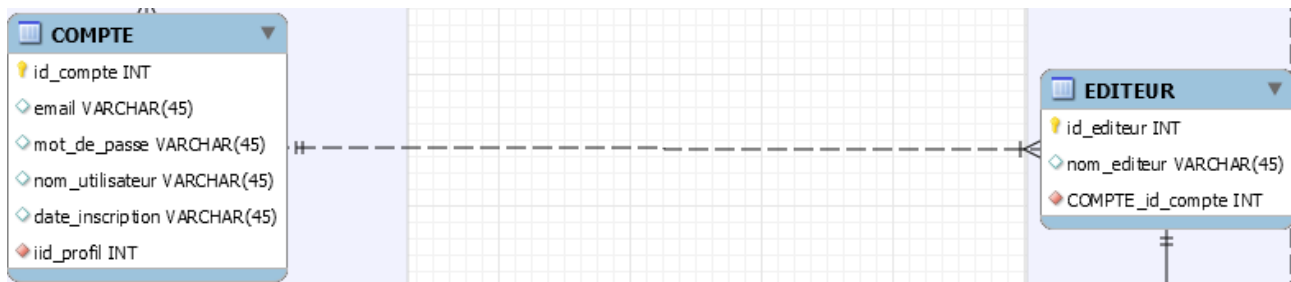
Pour un premier temps nous n'avons considéré que trois type des groupes pour représenter les autres groupes non cités dans l'exercice



CHAQUE UTILISATEUR APPARTIENT A UN GROUPE

Nous avons pris un exemple parmi plusieurs pour illustrer notre cas.

Ici nous avons utilisé la relation (1 : n), dans le but de traduire simplement qu'un ou plusieurs auront plusieurs rôles, mais un rôle appartiendra à un seul compte(utilisateur).

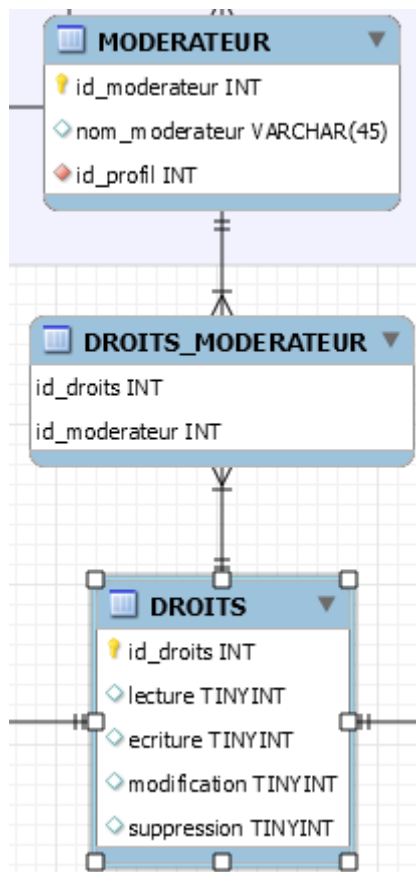


CHAQUE GROUPE DISPOSE DES DROITS SPECIFIQUES

Ici nous avons utilisé la relation (n : n), pour signifier que chaque groupe possède plusieurs droits et qu'en retour plusieurs droits peuvent s'appliquer sur plusieurs groupes, et cette relation (n : n) crée une table intermédiaire qui relie les 2 premières tables.

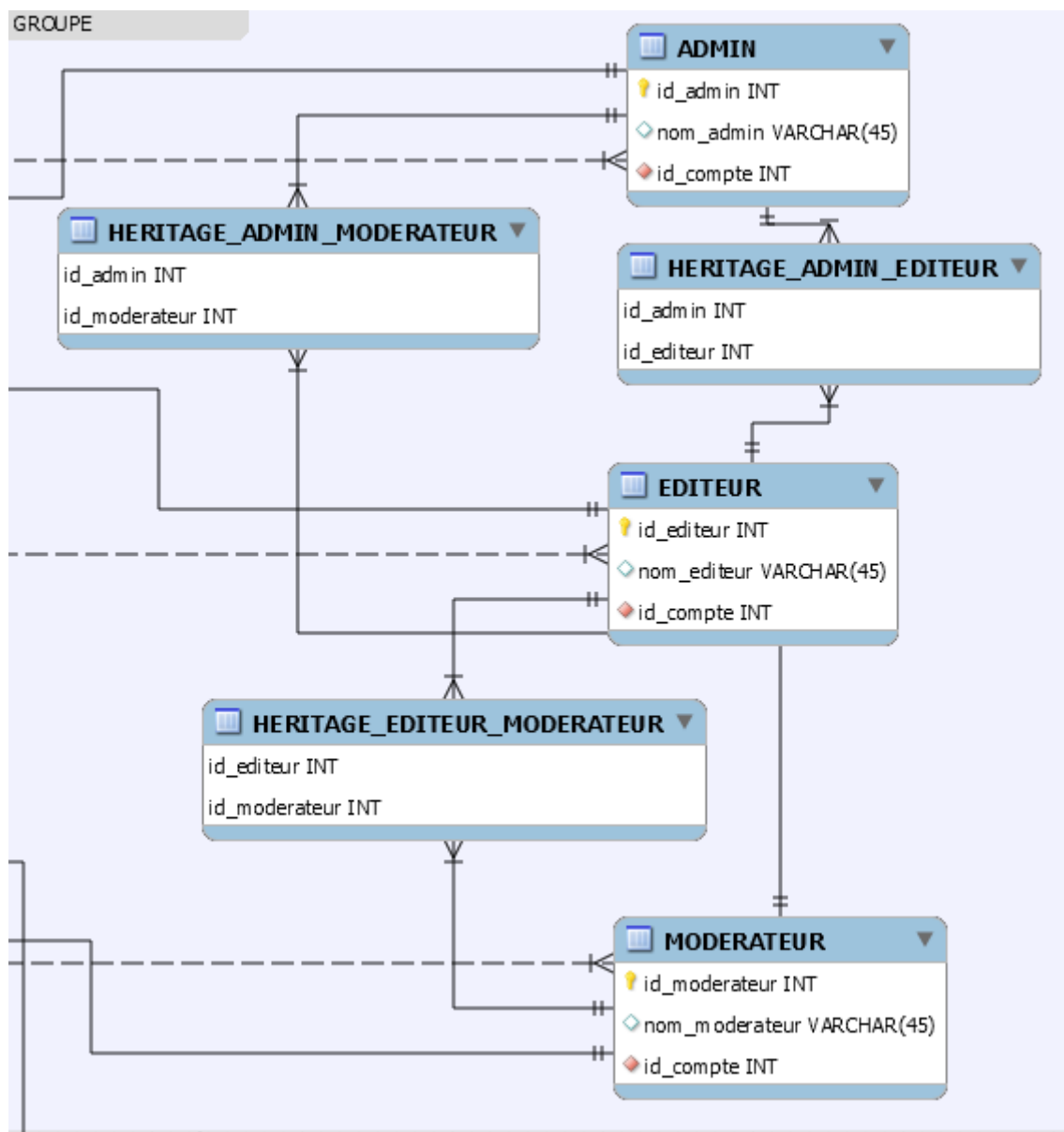
Dans la table DROITS nous avons utilisé le type booléen qui renvoie true(1) ou false(0), par ce que ces deux valeurs nous permettront de savoir si un utilisateur possède un droit ou pas.

True(1) pour l'utilisateur qui possède un droit et false(0) pour l'utilisateur qui n'en possède pas.



UN GROUPE PEUT HÉRITER DES DROITS D'UN AUTRE GROUPE

Nous avons utilisé avec 3 tables la relation (n:n) pour créer nos héritages des groupes.



Deuxième point : Création de nos tables pour le cas 19 minutes.

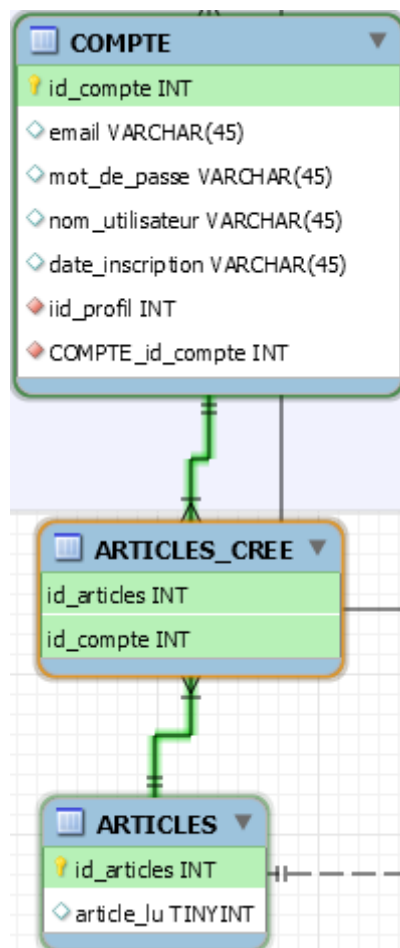
AVOIR UNE SECTION D'ARTICLE

Nous avons créé notre table ARTICLES.



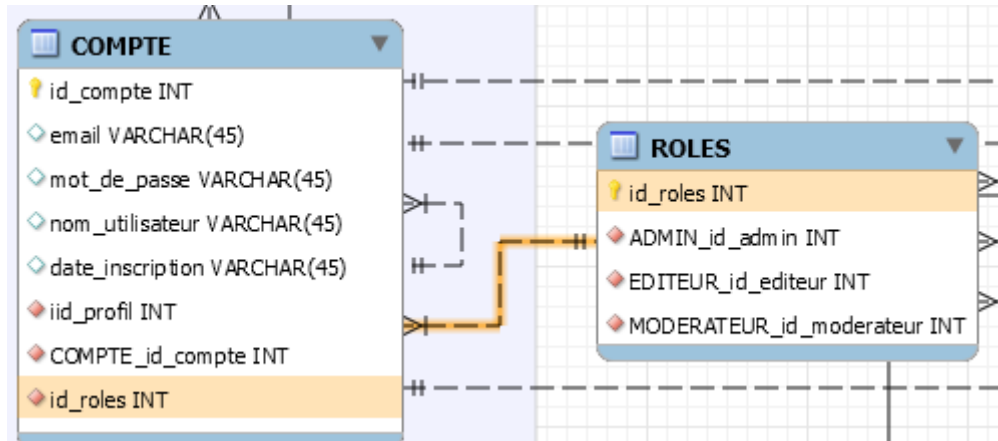
CHAQUE ARTICLE EST CRÉÉ PAR PLUSIEURS PERSONNES

Nous avons ici utilisé la relation (n: n) pour signifier que chaque article est créé par plusieurs personnes et qu'aussi plusieurs personnes peuvent créer plusieurs articles.



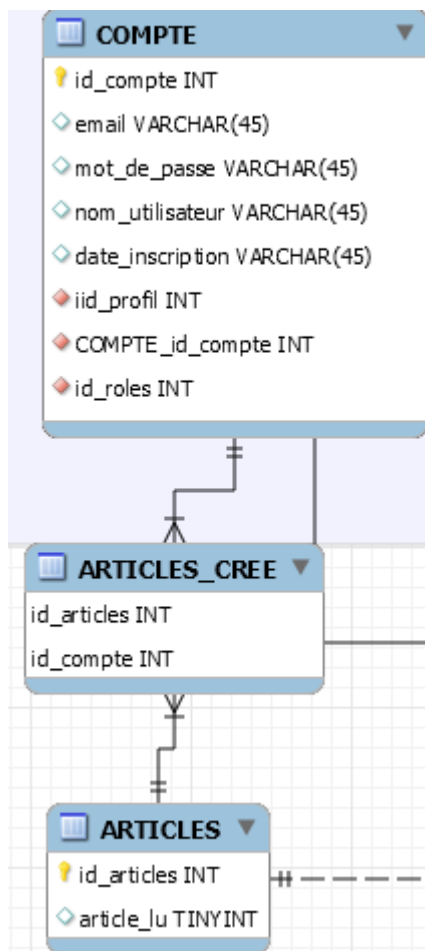
UNE PERSONNE PEUT AVOIR PLUSIEURS RÔLES

Nous avons en premier lieu utilisé la relation (1:n) entre les tables des rôles (admin, éditeur,...) et la table RÔLES, cette relation nous a permis de récupérer toutes les clés primaires de ces rôles et après cela nous avons utilisé la relation (1:n) entre la table RÔLES et la table COMPTE(Utilisateur)



LES UTILISATEURS DEVRONT SAVOIR S'ILS ONT LU L'ARTICLE OU S'ILS NE L'ONT PAS LU

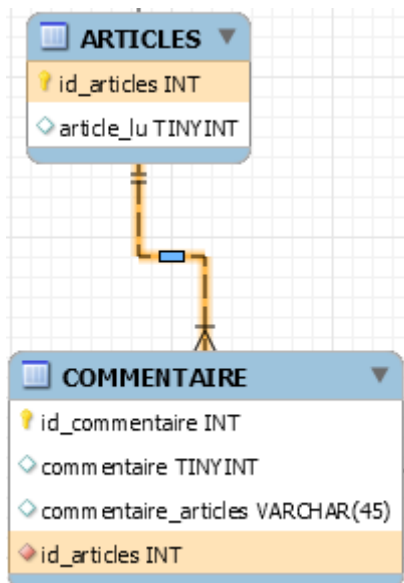
Dans la table ARTICLE, nous avons juste ajouté le champs `article_lu` ayant comme type booléen, ce champs aura soit (1) comme designer article lu et (0) pour designer article non lu.



CHAQUE ARTICLE POURRA OU PAS ÊTRE COMMENTÉ PAR DES UTILISATEURS

Nous avons utilisé la relation (1: n) pour signifier que chaque article pourra être connecté, et cette relation a transmise à la table COMMENTAIRE la clé primaire de la table ARTICLES.

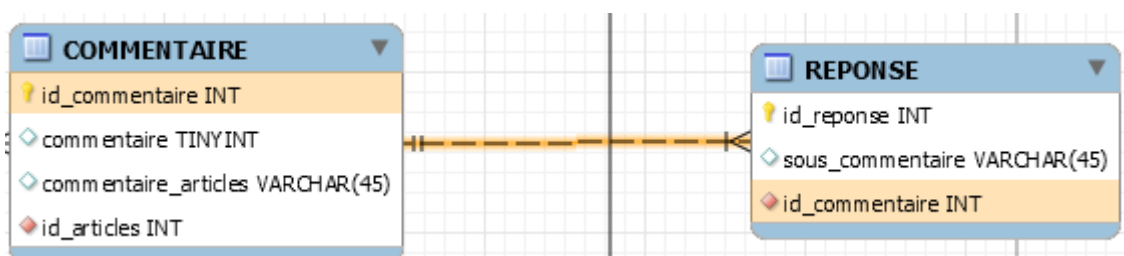
Dans la table COMMENTAIRE nous avons ajouté le champs commentaire ayant comme type booléen qui aura la valeur (1) pour signifier que l'article peut être commenté et (0) pour dire que l'article ne peut pas être commenté.



CHAQUE COMMENTAIRE POURRA AVOIR PLUSIEURS RÉPONSES, SOUS FORME DE SOUS COMMENTAIRES

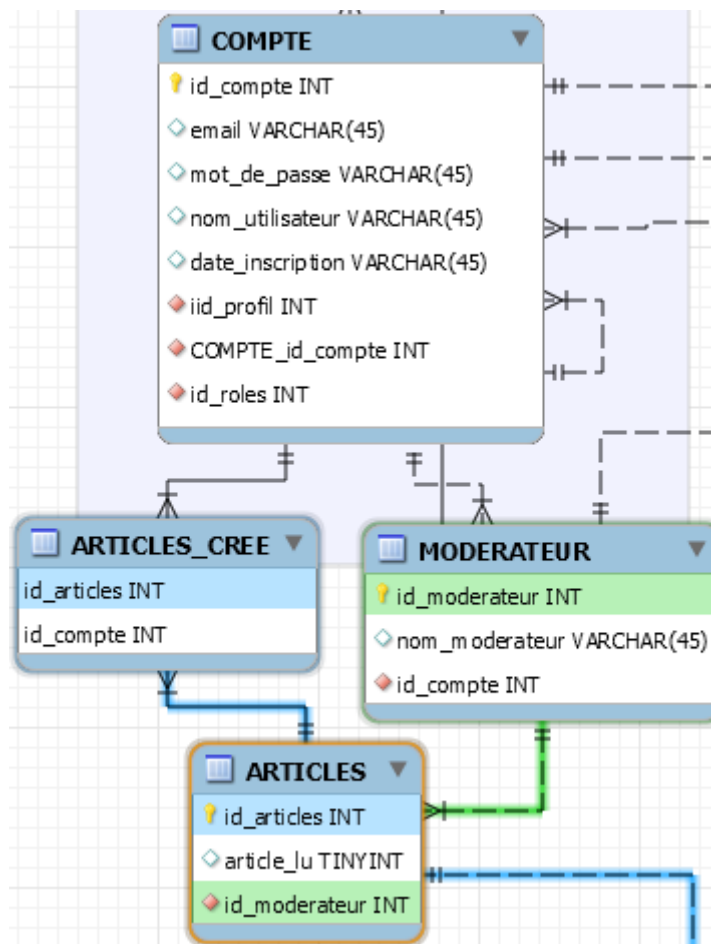
Nous avons utilisé ici la relation (1:n) pour signifier que chaque commentaire pourra avoir plusieurs réponses.

Dans la table REPONSE nous avons ajouté le champ sous_commentaire.



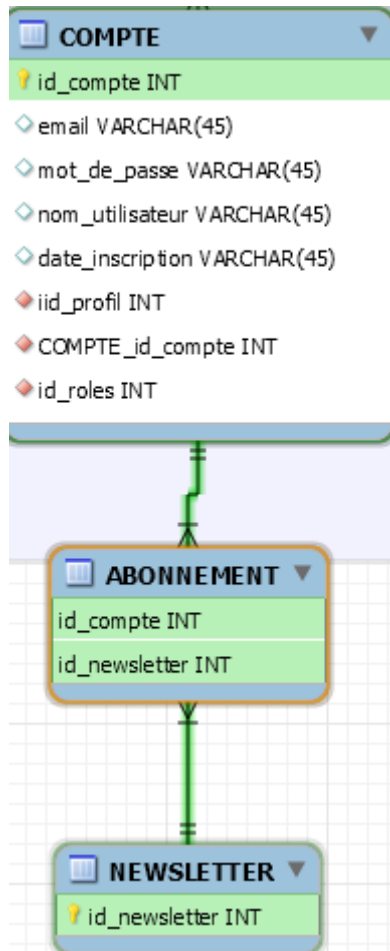
POUR CHAQUE ARTICLE, UN UTILISATEUR POURRA ÊTRE DÉSIGNÉ POUR FAIRE LA MODÉRATION, PAR DÉFAUT CE SERA LE CRÉATEUR DE L'ARTICLE.

Après avoir créé l'article par l'utilisateur, nous avons utilisé la table ARTICLE pour l'associer avec la table MODÉRATEUR grâce à la relation (1:n).



UN UTILISATEUR DEVRA S'ABONNER A UNE OU PLUSIEURS NEWSLETTER

Nous avons ici utilisé la relation (n: n) entre la table COMPTE et la table NEWSLETTER et cette relation nous a créé la table que nous avons renommé par ABONNEMENT.



UN UTILISATEUR POURRA S'ABONNER AUSSI A UN AUTRE UTILISATEUR POUR SAVOIR QUAND UN ARTICLE LE CONCERNANT EST PUBLIE.

Nous avons créé une relation (1:n) dans la table COMPTE et cette relation a créé une clé étrangère sur base de la clé primaire existant dans la table COMPTE.

