

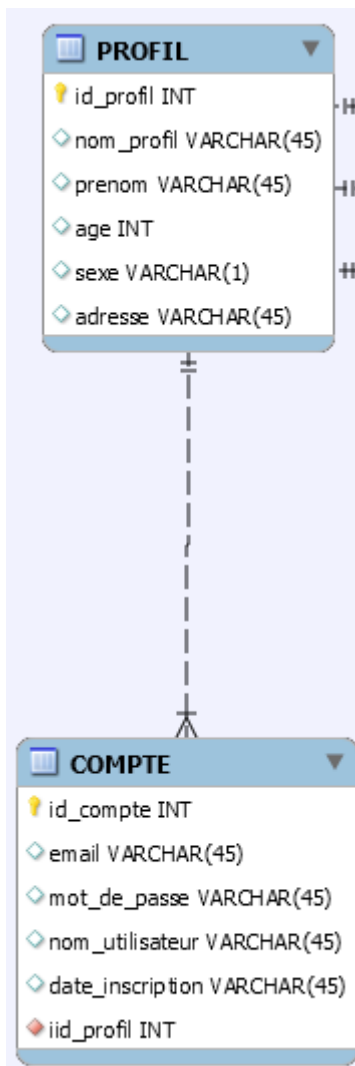
## EXPLICATION CAS : BON BILLET

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.

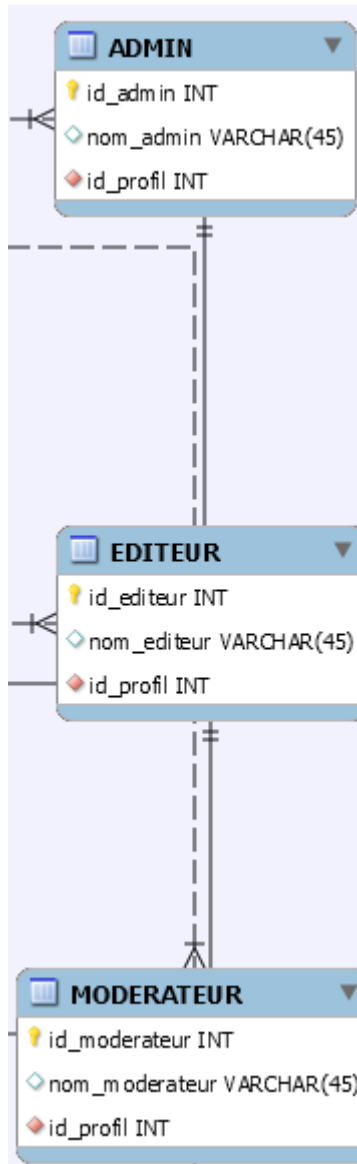
Nous avons définie ici la relation (1 : n) qui nous permet de dire qu'un ou plusieurs profils peuvent avoir un compte, mais un seul compte est attaché à un profil.

### CRÉATION DES TABLES POUR UTILISATEURS



## CRÉATION DES TABLES POUR LES GROUPES

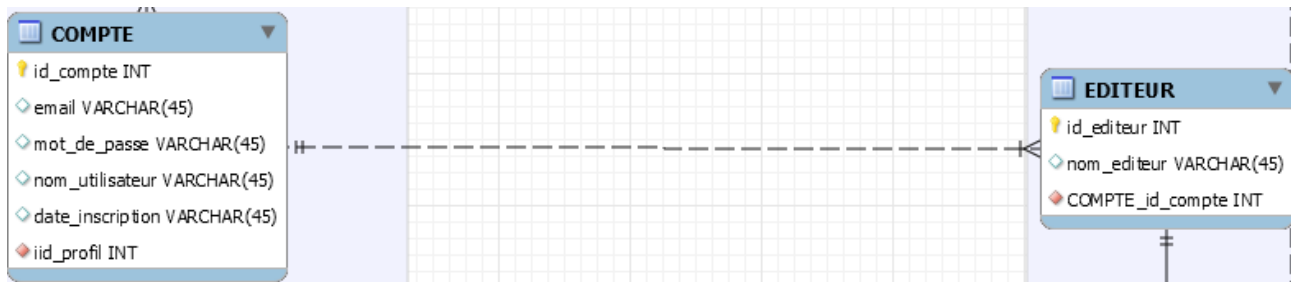
Pour un premier temps nous n'avons considéré que trois type des groupes pour représenter les autres groupes non cités dans l'exercice



## 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).

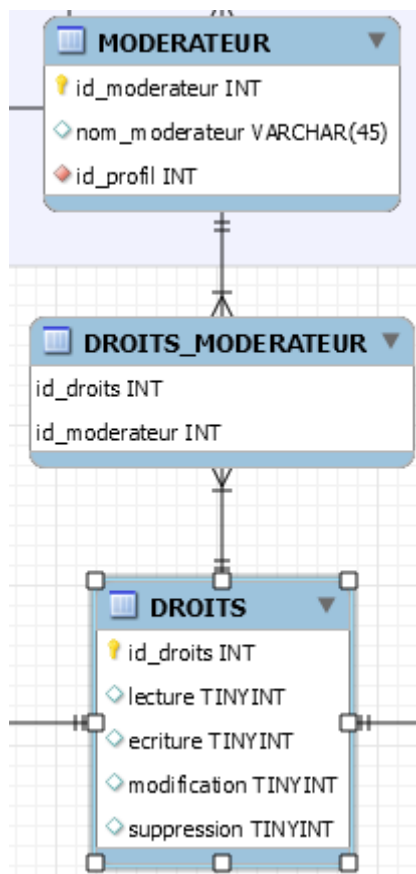


### CHAQUE GROUPE DISPOSE DES SPECIFIQUES

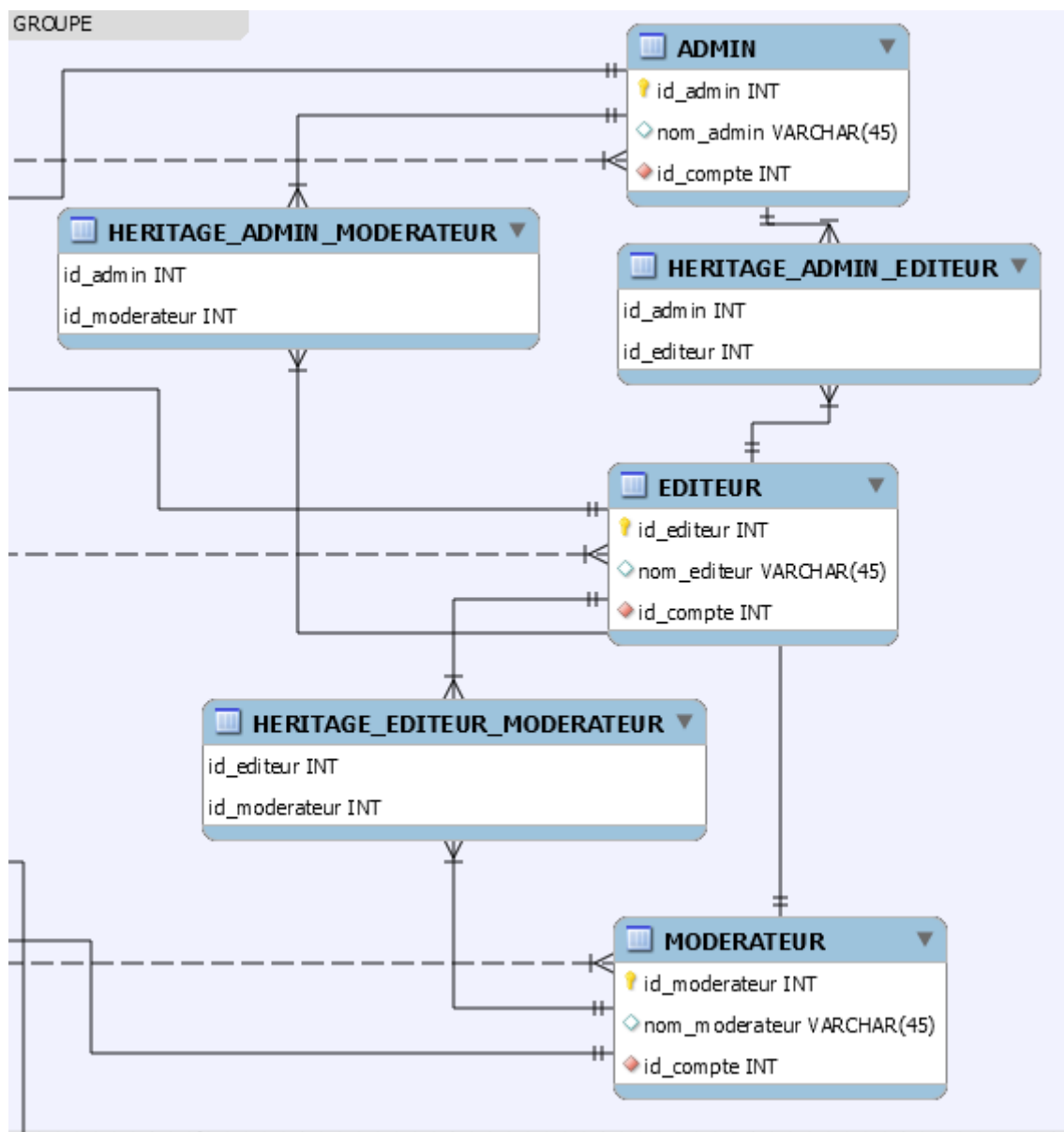
Ici nous avons utilisé la relation (n : n), pour signifier que chaque groupe possède plusieurs droits et qu'en retour plusieurs droits peuvent s'appliquer sur plusieurs groupes, et cette relation (n : n) crée une table intermédiaire qui relie les 2 premières tables.

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.



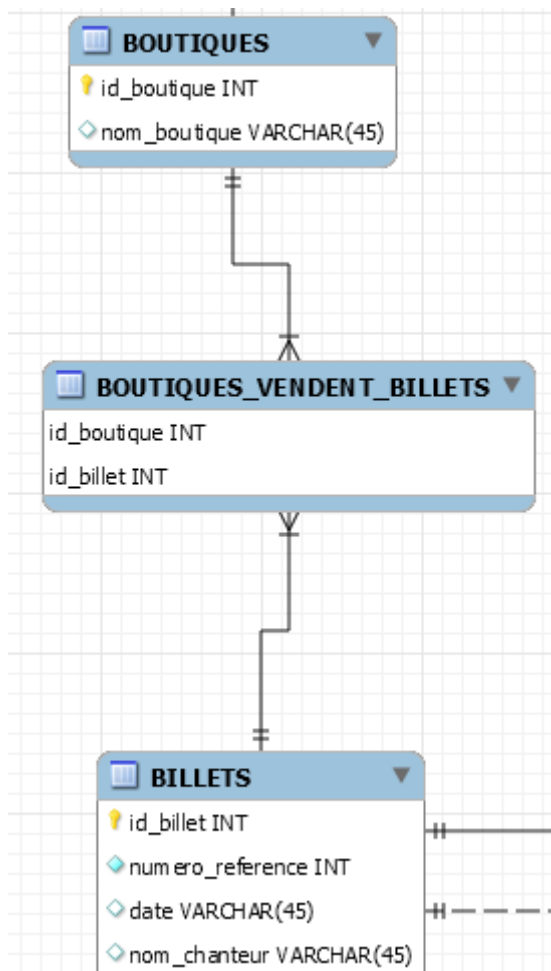
## UN GROUPE PEUT HÉRITER DES DROITS D'UN AUTRE GROUPE



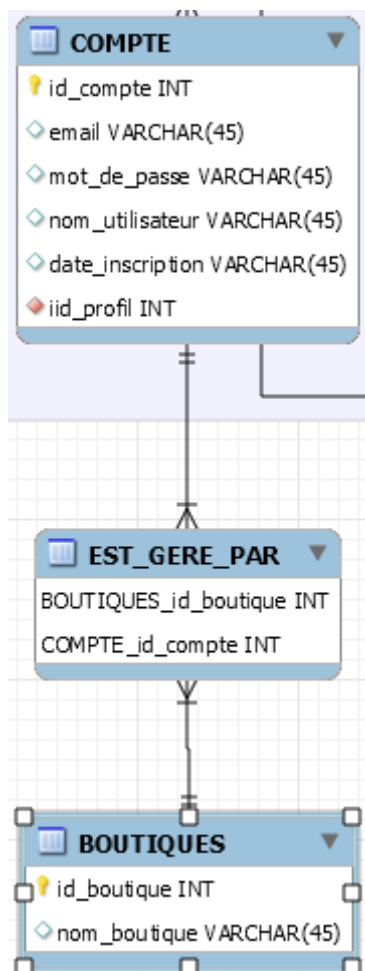
Deuxième Point : création des tables selon les cas...

LES BOUTIQUES SONT DES ENTITÉS QUI PERMETTENT DE VENDRE LES BILLETS

Nous avons créé deux tables et nous avons utilisé la relation (n : n) pour signifier qu'un ou plusieurs boutiques vendent plusieurs billets et qu'en retour un ou plusieurs billets peuvent être vendus par plusieurs boutiques.

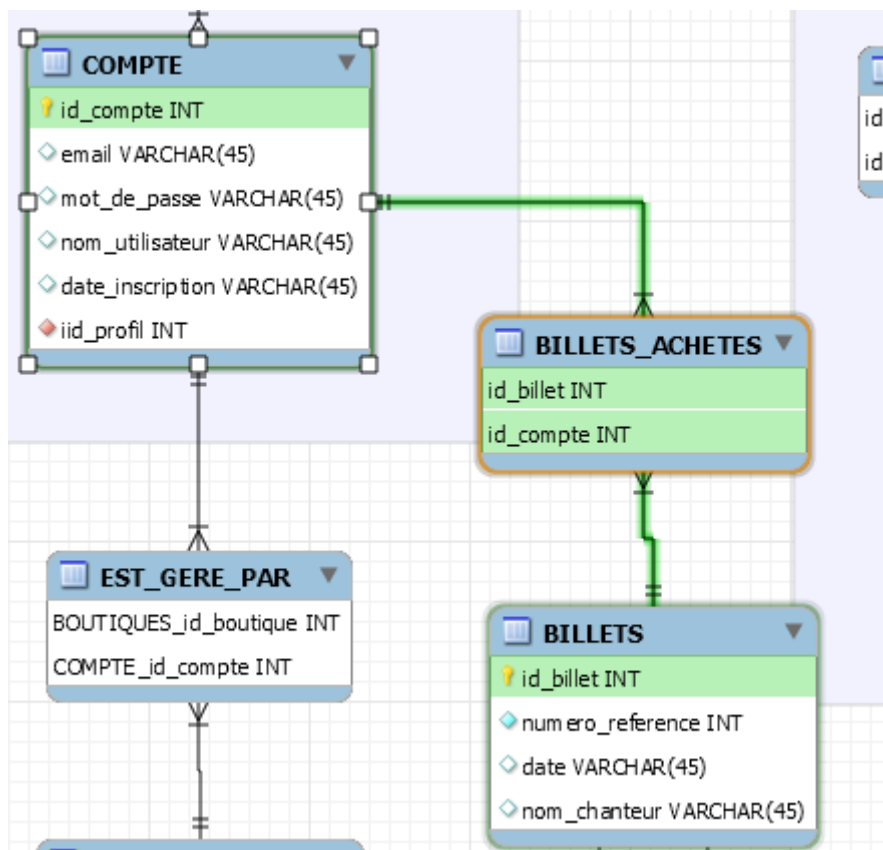


### UNE MÊME PERSONNE PEUT GÉRER PLUSIEURS BOUTIQUES



Nous avons ici utilisé la relation (n : n) pour signifier qu'une personne peut gérer plusieurs boutiques, et aussi une boutique peut être gérée par plusieurs personnes. Cette relation crée une table intermédiaire et nous retrouvons dans cette table intermédiaire les deux clés primaires de nos deux premières tables de départ.

CHAQUE BILLET APPARTIENT A UNE SEULE ET UNIQUE PERSONNE ET N'IMPORTE QUI POURRA ACHETER UN BILLET

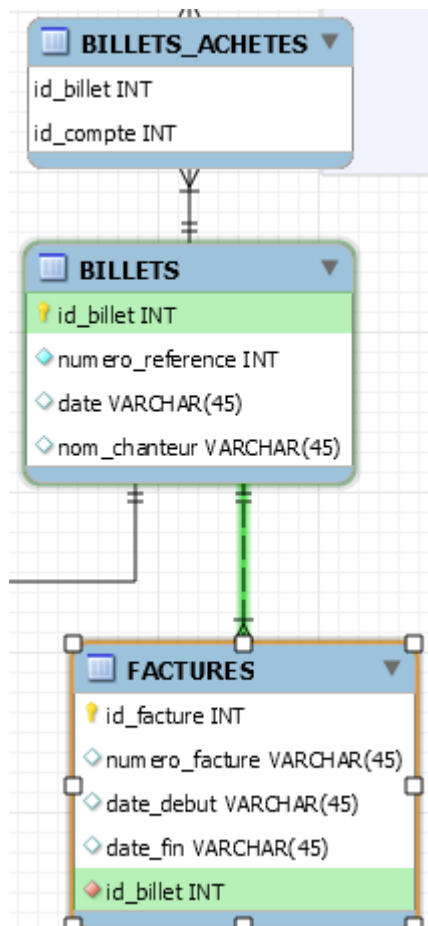


Nos deux tables Comptes(utilisateur) et BILLETS nous ont permis

d'utiliser la relation (n : n) pour signifier que chaque billets possèdent bien un compte et qu'un chaque compte possède aussi un plusieurs billets.

LE SITE DEVRA FOURNIR UNE FACTURE AINSI QUE LES REFERENCES DU BILLET

Nos deux tables BILLETS et Factures nous ont permis d'utiliser la relation (1 : n), et cette relation nous a permis de récupérer le numéro de référence qui est la clé primaire de la table Billet.



UNE FACTURE EST DÉFINIE PAR UN NUMÉRO UNIQUE QUI SE SUIT ET QUI SE RÉINITIALISE TOUS LES MOIS

Dans notre table Facture, nous avons créé une date début d'un billet et une date d'expiration et dépassé la date d'expiration, le billet devra se réinitialiser de manière automatique.

