

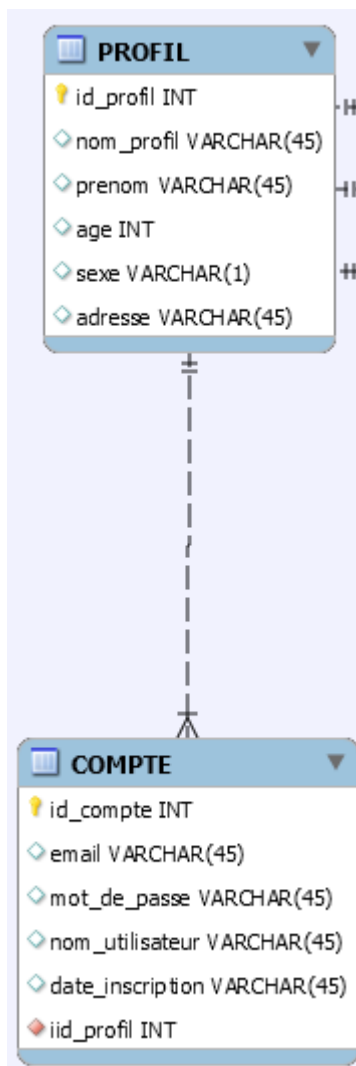
EXPLICATION CAS : E-SPORT

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.

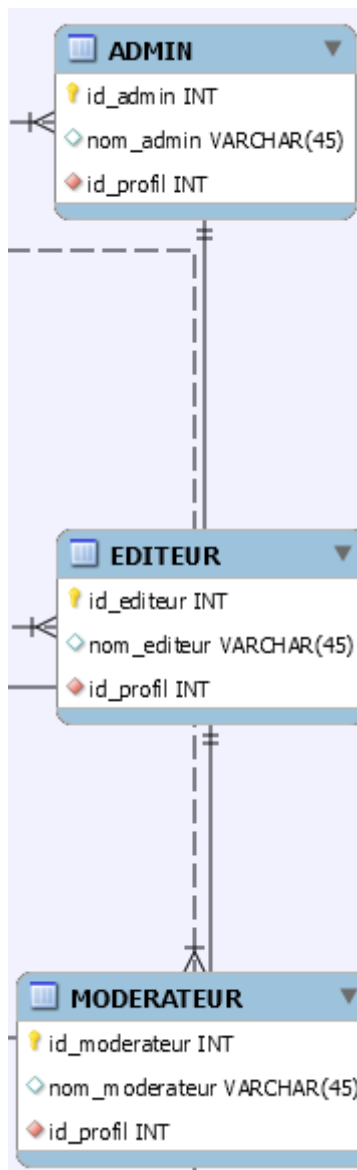
Nous avons définie ici la relation (1 : n) qui nous permet de dire qu'un ou plusieurs profils peuvent avoir un compte, mais un seul compte est attaché à un profil.

CRÉATION DES TABLES POUR UTILISATEURS



CRÉATION DES TABLES POUR LES GROUPE

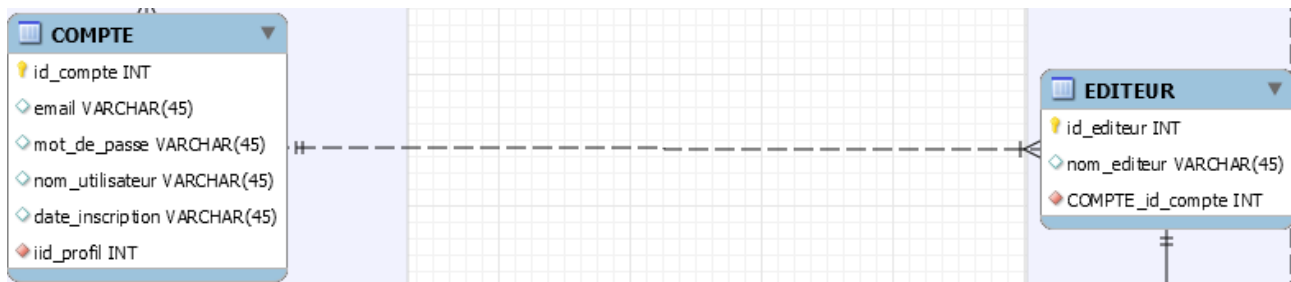
Pour un premier temps nous n'avons considéré que trois type des groupes pour représenter les autres groupes non cités dans l'exercice



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).

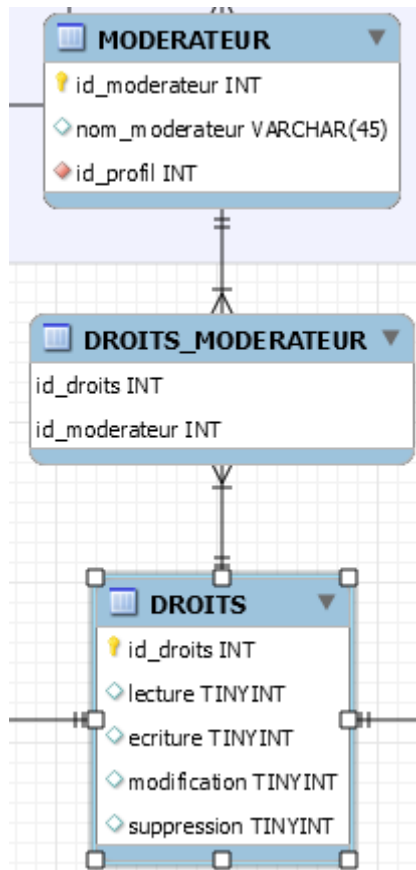


CHAQUE GROUPE DISPOSE DES DROITS SPECIFIQUES

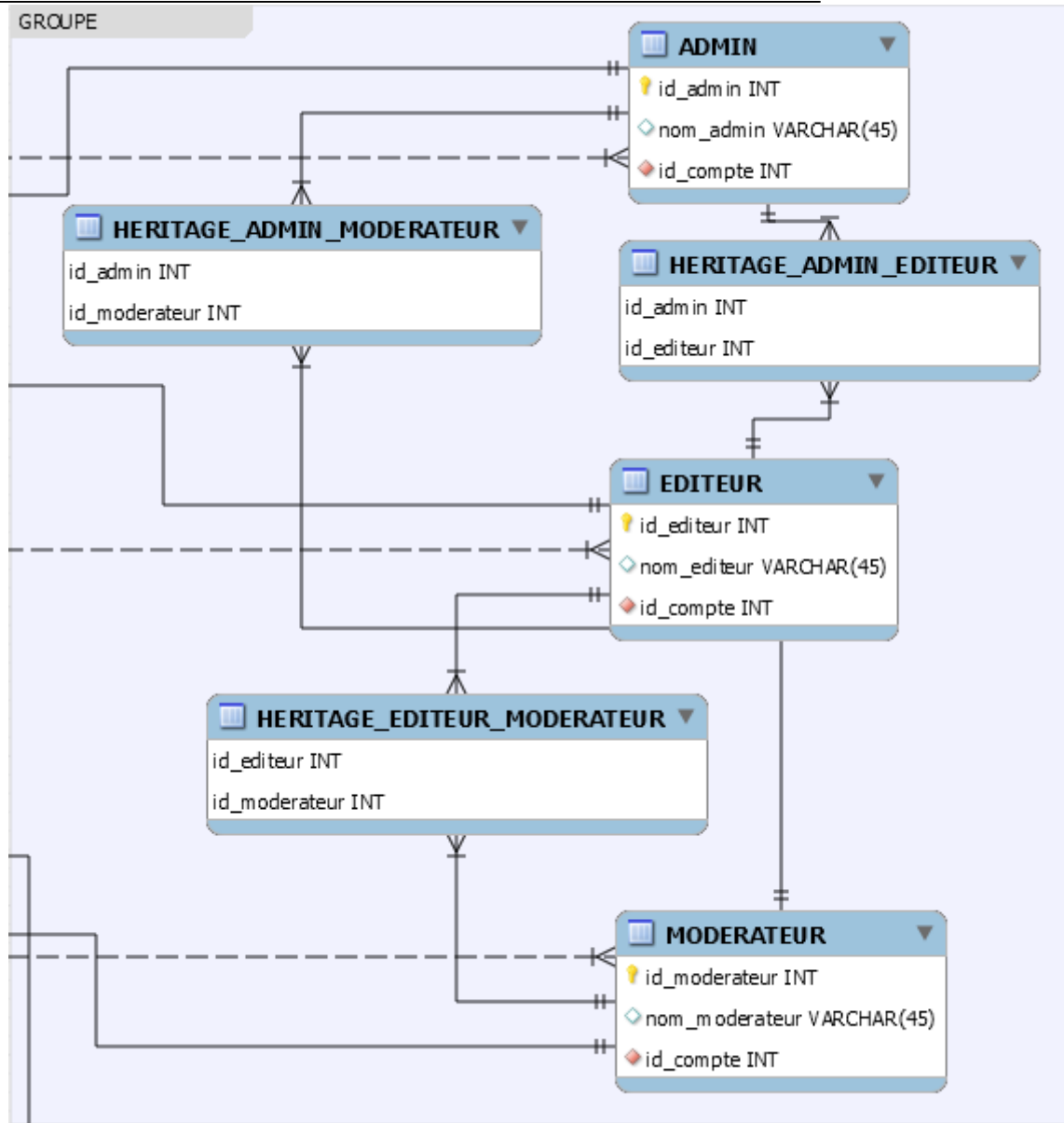
Ici nous avons utilisé la relation (n : n), pour signifier que chaque groupe possède plusieurs droits et qu'en retour plusieurs droits peuvent s'appliquer sur plusieurs groupes, et cette relation (n : n) crée une table intermédiaire qui relie les 2 premières tables.

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.

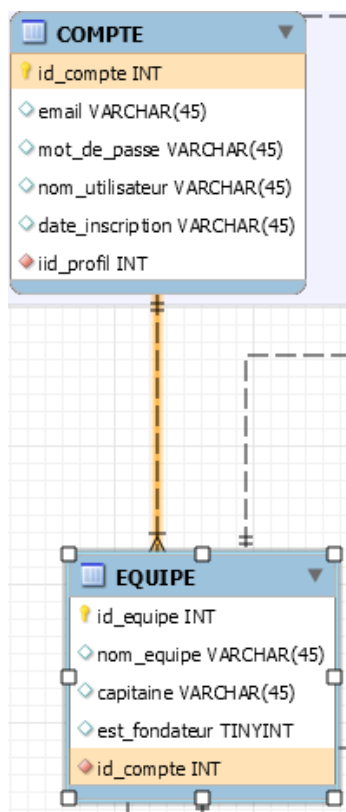
True(1) pour l'utilisateur qui possède un droit et false(0) pour l'utilisateur qui n'en possède pas.



UN GROUPE PEUT HÉRITER DES DROITS D'UN AUTRE GROUPE

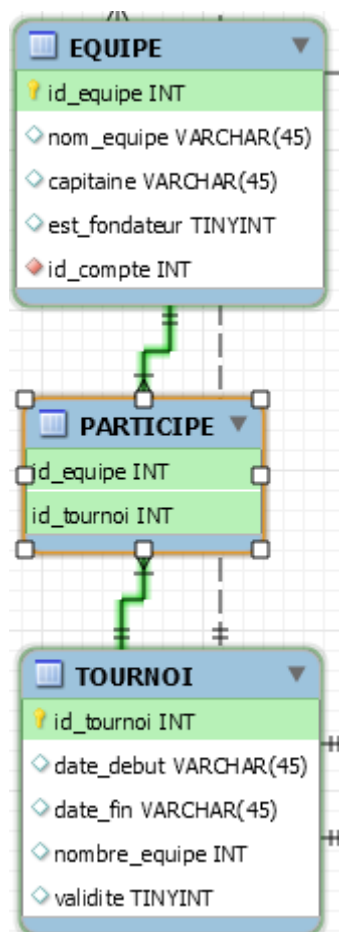


DEUXIÈME POINT : Création des tables pour le cas E-SPORT



LES ÉQUIPES SONT CONSTITUÉES D'UN OU PLUSIEURS UTILISATEURS

Grace à la relation (1 : n), nous voyons bien que la clé primaire de la table Compte(UTILISATEUR) devient une clé étrangère dans la table ÉQUIPE pour signifier que les équipes sont constituées d'un ou plusieurs utilisateurs.

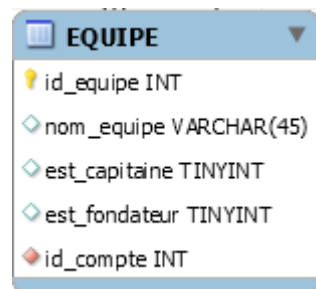


LES ÉQUIPES PEUVENT PARTICIPER A PLUSIEURS TOURNOIS

Nous avons utilisé la relation (n : n) et nous voyons bien qu'ici cette relation a créé une table intermédiaire que nous avons nommé participe et cette table récupère les deux clé primaires de nos tables de base et ces deux clés primaires deviennent des clés étrangères dans la table PARTICIPE.

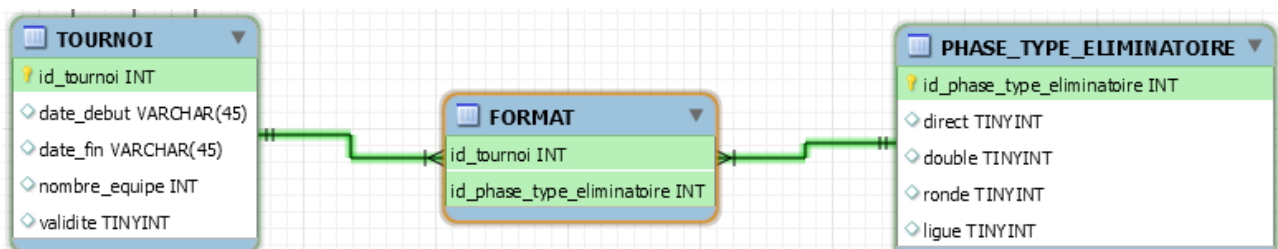
CHAQUE ÉQUIPE DISPOSE D'UN CAPITAN, QUI N'EST PAS FORCEMENT LE FONDATEUR

Dans la table ÉQUIPE, nous avons ajouter deux champs, le champs est_capitaine qui a un type booléen et qui permet de donner une valeur à chaque joueur qui sera soit (1), soit (0). (1) pour désigner le capitaine de l'équipe et (0) pour ceux qui ne seront pas capitaine. Et le champs est_fondateur qui a aussi un type booléen et qui permet de donner une valeur à utilisateur, s'il est fondateur l'utilisateur aura (1) comme valeur et sinon l'utilisateur aura (0) comme valeur.



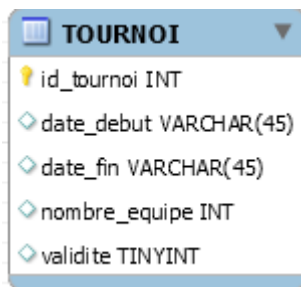
UN TOURNOI EST DÉFINI PAR SON FORMAT , UNE DATE DE DÉBUT ET UNE DATE DE FIN

Nous avons ici utilisé la relation (n : n) pour créer une table intermédiaire que nous avons renommé par FORMAT qui regroupe les deux clés primaires de nos deux tables de base. Et là nous pouvons bien voir que le tournoi a bel bien son format.



UN TOURNOI N'EST PAS VALIDE S'IL N'Y PAS ASSEZ D'ÉQUIPE DANS LE TOURNOI.

Nous avons ajouté deux champs dans notre table TOURNOI qui sont nombre_equipe qui a un type (INT) qui nous permettra de stocké le nombre d'équipe qui participe au tournoi et nous avons ajouté le champs validité qui a un type (BOOLEEN) qui aura soit la valeur (1) si le nombre d'équipe sont valide sinon (0).

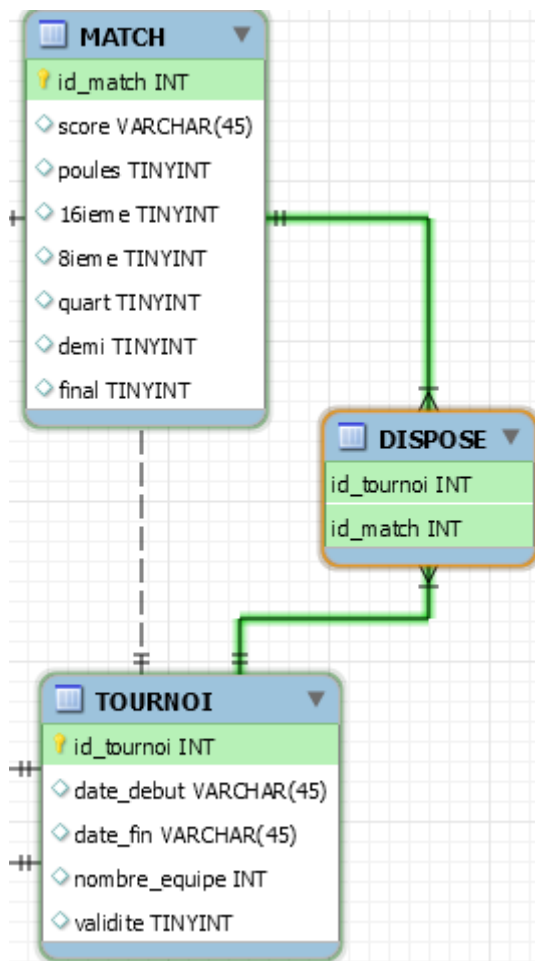


The image shows a screenshot of a database table structure for a table named 'TOURNOI'. The table has five columns: 'id_tournoi' (INT, primary key), 'date_debut' (VARCHAR(45)), 'date_fin' (VARCHAR(45)), 'nombre_equipe' (INT), and 'validite' (TINYINT). The columns are listed in a box with a blue header and a dropdown arrow.

id_tournoi	date_debut	date_fin	nombre_equipe	validite
INT	VARCHAR(45)	VARCHAR(45)	INT	TINYINT

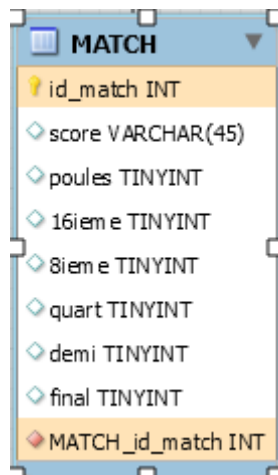
CHAQUE TOURNOI DISPOSE DE PLUSIEURS MATCHS

Nous avons utilisé ici la relation (n : n) pour signifier que chaque tournoi dispose de plusieurs matchs ainsi que plusieurs match peuvent se jouer dans plusieurs tournois.



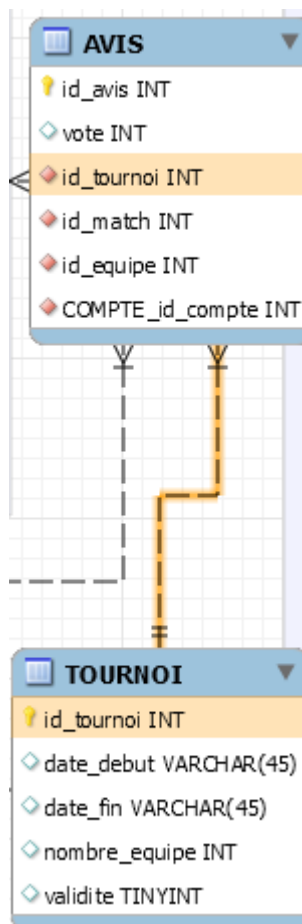
LES MATCHS SONT INTERCONNECTES ENTRE EUX

Nous avons créé la connexion de la table match avec elle-même et cette table a produit une clé étrangère sur la clé primaire existante.

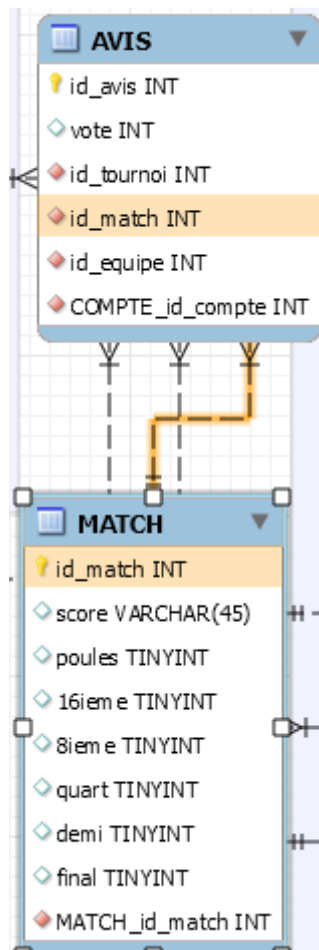


CHAQUE UTILISATEUR POURRA DONNER SON AVIS POUR UNE ÉQUIPE, UN MATCH, TOURNOI AVEC UN SYSTÈME DE VOTE DE 0 A 5 ÉTOILES.

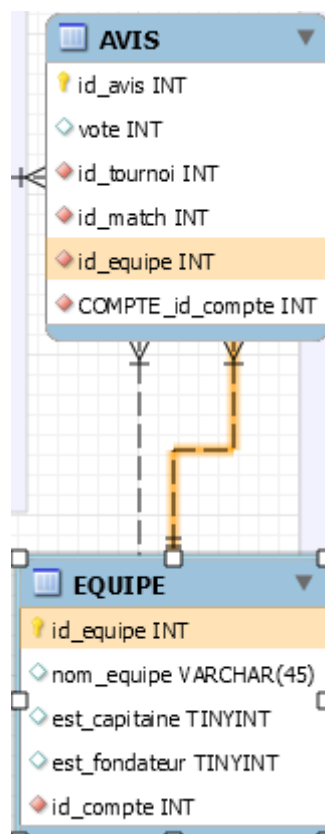
Nous avons récolté les données de la table AVIS en quatre étapes
la première étape: nous utilisons la relation (1: n)
Nous utilisons la table AVIS et la table TOURNOI



Deuxième étape: nous utilisons toujours la relation(1:n)
Nous utilisons la table AVIS et la table match

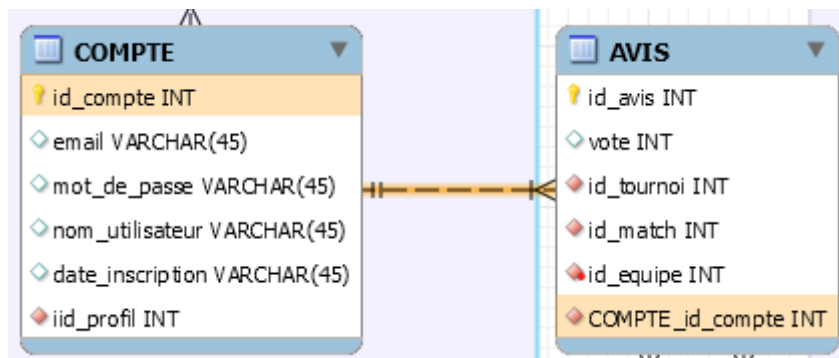


Troisième étape: nous utilisons toujours la relation (1:n)
Nous utilisons la table AVIS et la table ÉQUIPE



Quatrième étape: nous utilisons toujours la relation (1:n)

Nous utilisons la table AVIS et la table COMPTE (UTILISATEUR)



CONCLUSION

Nous voyons bien que toutes les clés primaires sont maintenant dans la table AVIS comme des clés étrangères.

