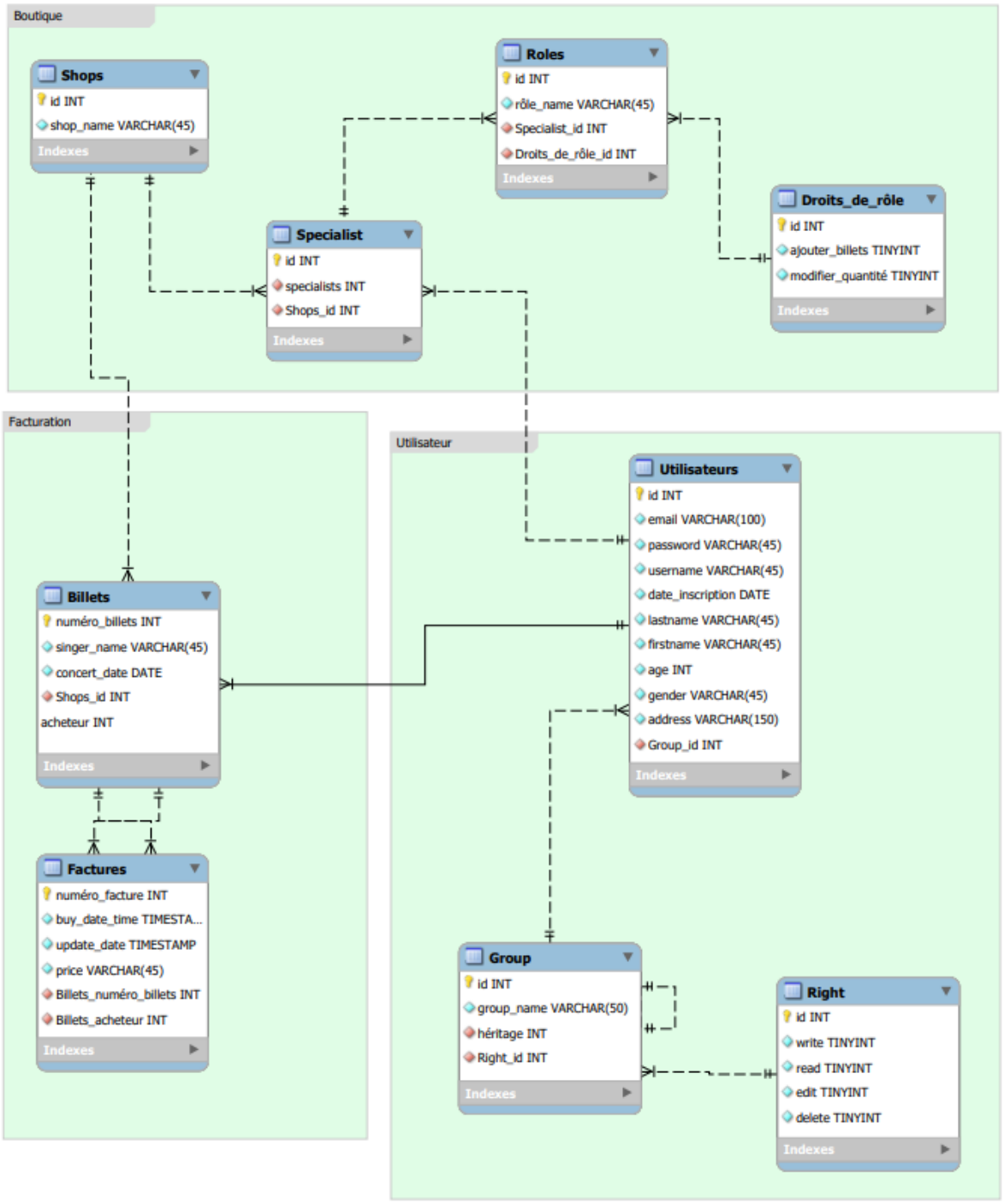
Le Bon Billet

Introduction

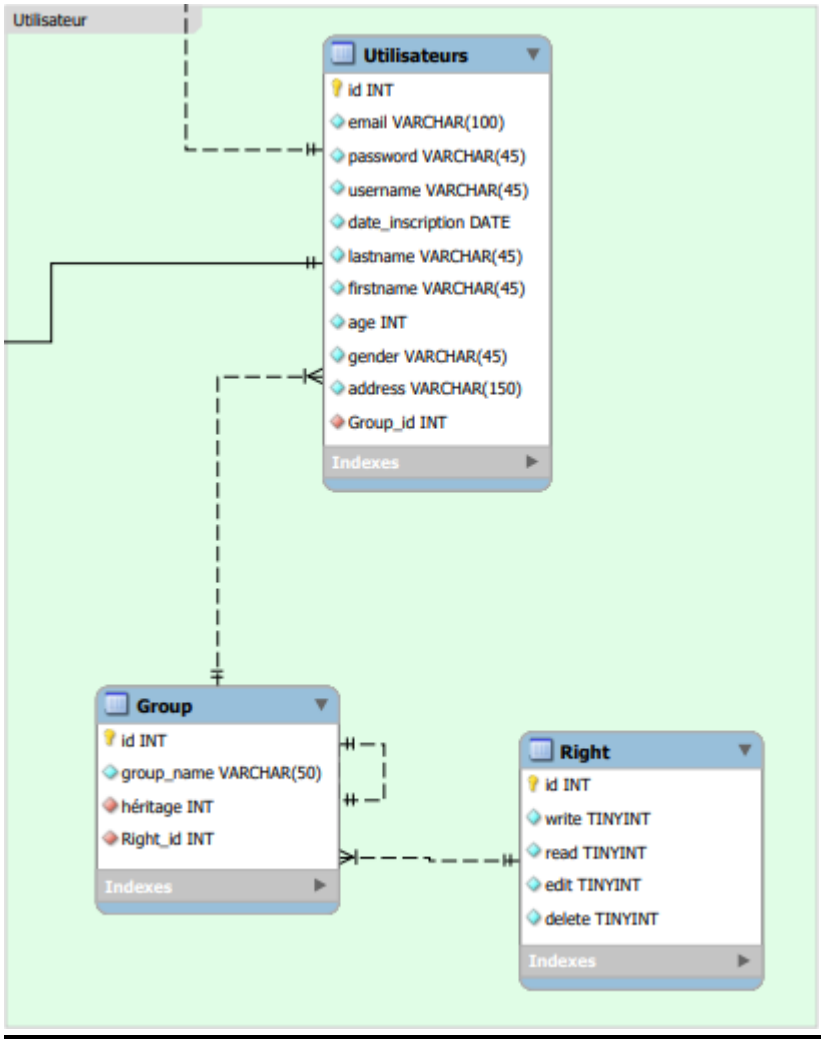
Nous avons élaboré, ci-dessous, le schéma de la base de données de votre future plateforme de tournois e-sportifs respectant tous les critères demandés. Cette base de données vous permettra de réaliser tout type de tournois.



Ce schéma est divisé en 3 parties distinctes :

- Une première partie, contenant les besoins communs à la plupart des sites internet, regroupant toutes les informations concernant un utilisateur que ce soient ses droits ou ses informations personnelles.
- Une seconde partie représentant le système de vote permettant à chaque utilisateur de donner son avis sur une équipe, un match ou bien un tournoi.
- Une troisième partie qui va regrouper l'ensemble des informations concernant le tournoi en lui-même que nous détaillerons par la suite.

Les tables de l'Utilisateur



Constituée de 3 tables afin de faciliter la compréhension du schéma, pour que ce dernier soit plus clair, cette table regroupe, comme il a été dit précédemment, toutes les informations de l'utilisateur de ses droits à ses informations personnelles.

Nous avons tout d'abord une première table qui se nomme « Utilisateurs » contenant toutes les informations personnelles de l'utilisateur tels que l'email, son nom d'utilisateur, son mot de passe etc … Nous avons utilisé des

VARCHAR pour les données utilisant tout type de caractères, puis d'autres types particuliers pour chacune des données nécessitant tel ou tel type.

Puis, une seconde table, étroitement reliée avec la troisième, qui se nomme « Group » contenant le nom d'un groupe définissant un rôle associé à des droits d'utilisations sur le site, ces derniers se trouvant dans la dernière table nommée « Right ».

Les relations entre les tables

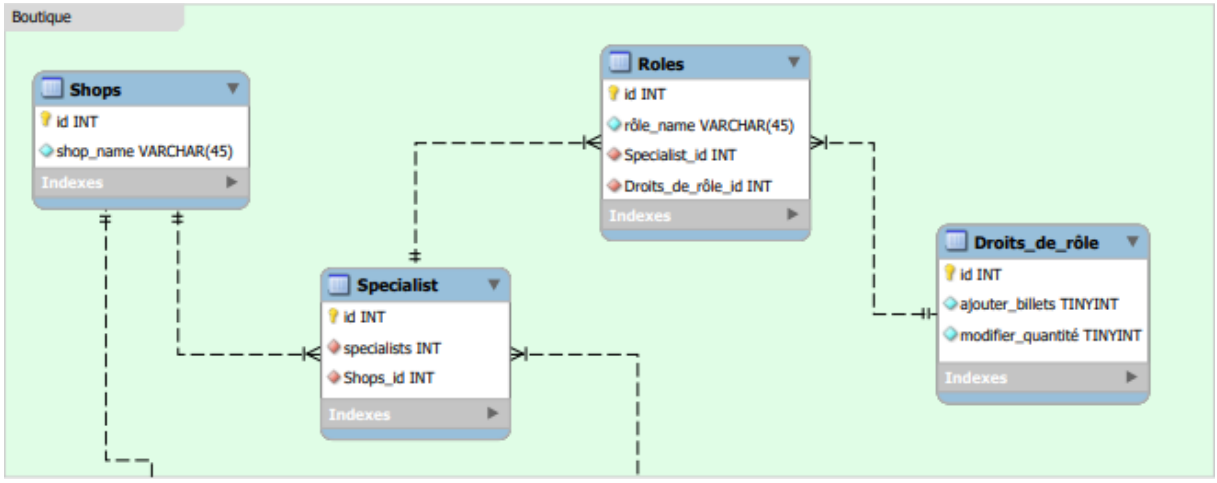
Voyons maintenant les relations entre ces tables.

Nous avons fait un lien dit « one-to-many » (noté 1 : n) entre la table Utilisateur et la table Group afin que chacun des utilisateurs soit associé à un groupe, et pour que plusieurs utilisateurs puissent avoir le même rang.

Nous avons ensuite fait un lien dit « one-to-one » (noté 1 : 1) au sein même de la table Group afin de permettre à un groupe d'hériter des droits d'un autre groupe.

Enfin, nous avons fait une relation 1 : n entre la table Right et la table Group afin de donner des droits différents à chacun des groupes, et, pour que ces derniers n'aient qu'une seule série de droits, c'est-à-dire qu'une seule ligne de chacun des quatre droits soit associée à un seul groupe.

Les tables de la Boutique



Constitué de 4 tables, les tables représentant la boutique regroupent chacune quelques informations à son propos.

Nous pouvons trouver dans une première table nommée « Shops » l'identifiant de chacune des boutiques ainsi que leurs noms, puis dans une seconde table nommée « Specialist » l'identifiant des utilisateurs qui seront désignés comme les spécialistes qui gèreront la boutique.

Associée directement à cette table, une troisième table qui se nomme « Roles » elle-même liée à une quatrième table qui se nomme « Droits_de_rôle ».

Dans ces deux tables nous allons pouvoir retrouver les mêmes fonctionnalités que dans les tables de l'Utilisateur concernant les droits d'utilisation avec deux nouveaux droits : « Ajouter des billets » et « modifier la quantité ».

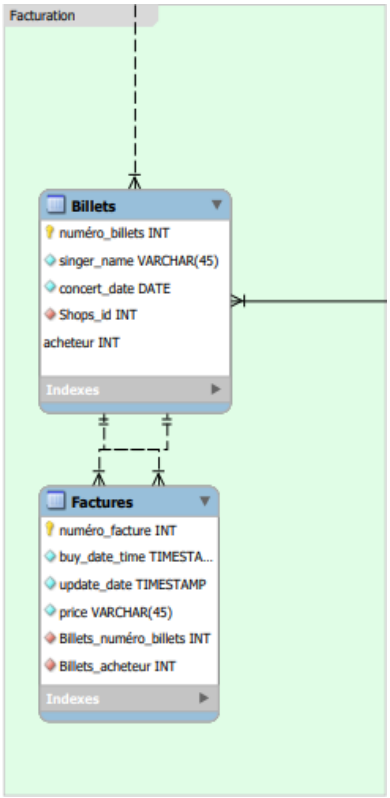
Nous avons créé ces tables dans le but de créer des droits distincts des utilisateurs qui seront spécifiques aux spécialistes pour plus de clarté dans notre base de données.

Nous avons donc une relation 1 : n entre la table Shops et la table Specialist puisqu'une boutique pourra être composée de plusieurs gérants.

Une relation 1 : n entre la table Specialist et la table Roles, dans ce sens-là, puisque les spécialistes auront des rôles différents et plus ou moins de droits selon leur rang au sein de la boutique.

La dernière relation étant complètement similaire à celle des tables Utilisateurs, aucun commentaire supplémentaire ne sera fait à son sujet.

Les tables de la Facturation



Nous allons maintenant voir comment se traitent les informations au niveau de la facturation.

Dans ce layer, nous pouvons voir qu’il y a 2 tables.

Dans une première table nommée « Billets » regroupant les informations d’un billet tels que le nom de l’artiste, la date du concert ainsi que l’identifiant de la boutique où a été achetée le billet sous la forme d’une clé étrangère et l’identifiant des acheteurs afin de récupérer leurs noms et éventuellement y inscrire leurs noms.

Dans la seconde table nommée « Factures » se trouvent les informations que l’on retrouve sur une facture tel que la date d’achat, le prix et le numéro du billet qui sera réinitialisé tous les mois ainsi que l’identifiant du billet de l’acheteur.

Glossaire

1:1 La relation one-to-one est définie comme une relation entre deux tables où les deux tables sont associées en utilisant la primary key unique et la foreign key contrainte . Un enregistrement de la table A se rapporte seulement à un enregistrement de la table B.

1:n La relation one-to-many est définie comme une relation entre deux tables, une ligne d'une table peut avoir plusieurs lignes correspondantes dans une autre table. Cette relation peut être créée à l'aide de la relation -clé étrangère. Un enregistrement de la table A se rapporte à un ou plusieurs enregistrements de la table B.

n:m La relation many-to-many est définie comme une relation entre deux tables où les deux tables sont associées. Un enregistrement de la table A se rapporte à un ou plusieurs enregistrements de la table B et un enregistrement de la table B se rapporte à un ou plusieurs enregistrements de la table A. Une relation n:n peut donc être considéré comme deux relation 1:n reliées par une table intermédiaire qu'on appelle "table de jonction". Cette table est utilisée pour relier les deux autres tables. Pour ce faire, deux champs font référence à la clé primaire de chacune des deux autres tables.

Identifying La relation "identifiante" existe lorsque la primary key de l'entité parent est incluse dans la primary key de l'entité enfant.

Non-identifying La relation "non-identifiante" existe lorsque la primary key de l'entité parent est incluse dans l'entité enfant mais ne fait pas partie de la primary key de l'entité enfant.

VARCHAR (taille) Une chaîne de variable peut contenir des lettres, des chiffres et des caractères spéciaux). Le paramètre spécifie la longueur maximale de la colonne en caractères - peut aller de 0 à 65535.

TEXT (taille) Contient une chaîne d'une longueur maximale de 65 535 octets.

ENUM (val1, val2, val3, ...) Un objet chaîne ne peut avoir qu'une seule valeur en choisant parmi une liste de valeurs possibles. Une liste ENUM peut répertorier jusqu'à 65 535 valeurs dans. Si une valeur est insérée et elle ne figure pas dans la liste, une valeur vide sera insérée. Les valeurs sont triées dans l'ordre où on les entre.

BOOLEAN Zéro est considéré comme faux, les valeurs autres que zéro sont considérées comme vraies.

TINYINT (taille) Un très petit nombre entier. Ce type de variable représente au type BOOLEAN.

DATE Une date format: YYYY-MM-DD.

DATETIME Une combinaison de date et heure. Format: YYYY-MM-JJ hh: mm: ss.

TIMESTAMPS Le timestamp représente le nombre de seconde qui s'est écoulé depuis le 1er janvier 1970 (début de l'heure UNIX). On utilise pour identifier les transactions.