

Projet Base de Données NoiseBook

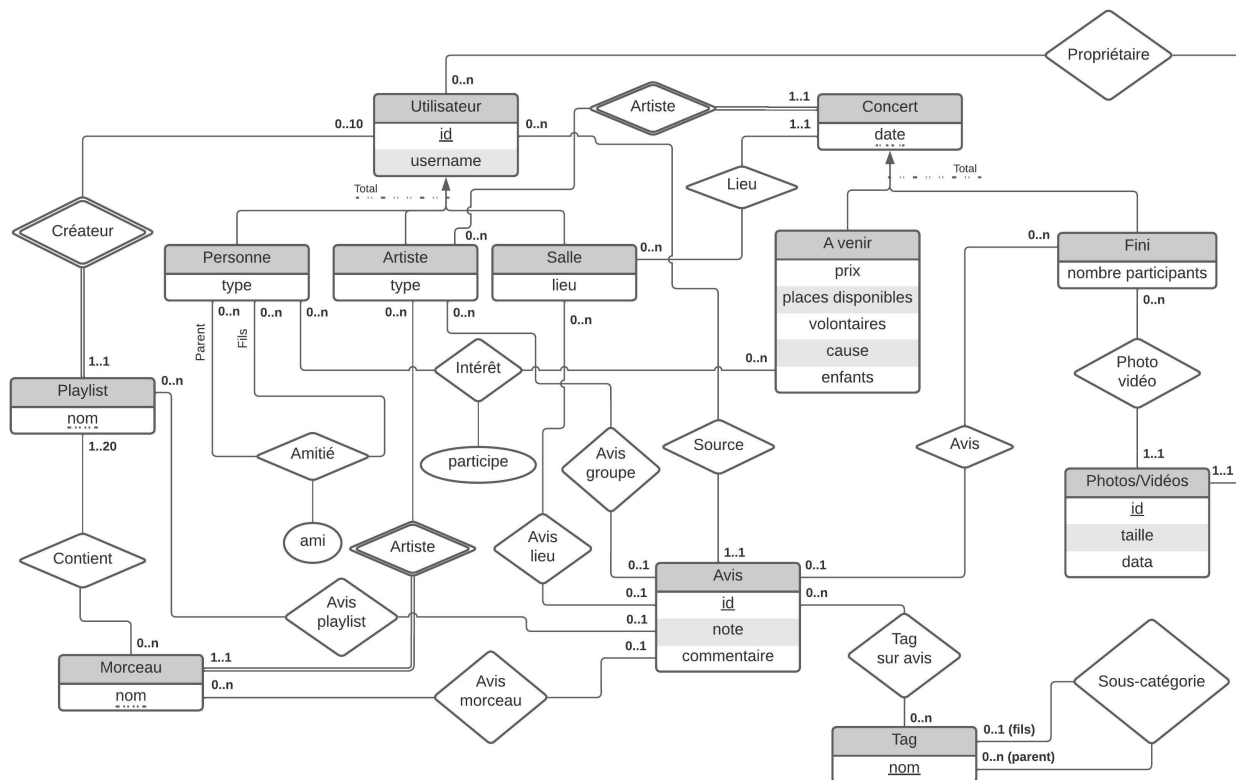
DEBBAH Martin
ZHOU Alexandre

I. Le diagramme entité/association

Nous avons commencé par réfléchir au schéma entité/association. Après avoir lu le projet, nous avons défini les différentes entités qui allaient constituer notre base de données.

Diagramme E/A NoiseBook

Martin DEBBAH | Alexandre ZHOU



- Utilisateur:

Dans notre implémentation, nous retrouvons trois types d'utilisateurs. Les personnes, qui sont représentés des personnes lambda. Les artistes, qui représentent des groupes ou des chanteurs. Et les salles, qui représentent les entités pouvant accueillir des artistes en concert. Chacun de ces types hérite d'une entité mère ayant comme attributs un entier id pour leur identifiant unique et une chaîne de caractères username pour leur nom. L'entité utilisateur est associée à:

- l'entité avis pour connaître l'utilisateur source d'un avis
- l'entité photos/vidéos pour connaître l'auteur d'une photo ou d'une vidéo
- l'entité playlist pour connaître le créateur de la playlist

Les personnes sont, en plus, associées à l'entité personne pour définir les relations d'amitié et à l'entité concert pour les relations d'intérêt envers un concert.

Les artistes sont associés aux entités concert et morceau pour définir l'artiste d'un concert ou d'un morceau, mais également à l'entité avis puisqu'un avis peut être adressé à un artiste.

Les salles ont un attribut supplémentaire lieu sous forme de chaîne de caractères, représentant la ville où elles se situent. Elles sont également associées à l'entité concert pour connaître la localisation de celui-ci et à l'entité avis pour les avis sur les lieux.

- Concert:

Il existe deux types de concerts, les concerts à venir et les concerts finis. Tous deux héritent d'une entité faible ayant pour discriminant une date. L'entité identifiante de concert et l'entité artiste. L'entité concert est également associée à l'entité salle.

L'entité concert à venir comporte les attributs prix, places disponibles, volontaires, cause et enfants. Il y a une association entre personne et concert à venir pour montrer l'intérêt d'une personne pour ce concert.

L'entité concert fini, quant à elle, comporte l'attribut nombre de participants. Elle a deux associations, une avec photos/vidéos pour avoir les photos et les vidéos de ce concert, et l'autre avec l'entité avis.

- Photos/vidéos:

Nous avons une entité photos/vidéos qui regroupe différentes informations en lien avec les photos et les vidéos d'un concert. Elle est associée à un utilisateur et un concert fini.

- Playlist:

L'entité playlist est une entité faible. Son discriminant est l'attribut nom et l'entité identifiante est utilisateur. Elle est associée à l'entité morceau pour définir les différents morceaux composants la playlist, ainsi qu'à l'entité avis.

- Morceau:

L'entité morceau est également une entité faible. Son discriminant est l'attribut nom et l'entité identifiante est artiste. Elle est associée à l'entité playlist et à l'entité avis.

- Tag:

L'entité tag représente à la fois le genre de musique, mais aussi tous les mots-clés que l'on peut utiliser en tant que tag. Elle a un attribut nom permettant de l'identifier. Elle a une association avec elle-même pour pouvoir faire un système de sous-catégorie.

- Avis:

Finalement, l'entité avis représente les différents avis pouvant être effectués sur les autres entités du schéma. Les attributs de cette entité sont un identifiant, une note et un commentaire. Elle est associée à l'entité utilisateur pour connaître l'auteur de l'avis. Elle est également associée aux entités concert fini, artiste, lieu, morceau, playlist pour connaître la cible de l'avis. Une association entre avis et tag permet de savoir quelles entités sont concernées par les différents tags.

II. Le schéma relationnel

Utilisateur(user_id, user_name, lieu, user_type)
amitie(user_id, utilisateur_2, ami)
Concerts_a_venir(artiste, concert_date, lieu, prix, places, volontaires, cause, enfants, extérieur)
Concerts_finis(artiste, concert_date, lieu, nb_participants)
Interet(user_id, artiste, concert_date, participe)
Photos_videos(id, user_id, taille, artiste, concert_date)
Tags(nom, sous_genre_de)
Morceaux(artiste, nom)
Playlist(user_id, nom)
morceau_de_playlist(user_morceau, nom_morceau, user_playlist, nom_playlist)
avis(id_avis, user_id, note, com)
avis_morceau(id_avis, user_morceau, nom_morceau)
avis_playlist(id_avis, user_playlist, nom_playlist)
avis_concert_fini(id_avis, artiste, concert_date)
avis_lieu(id_avis, lieu)
avis_artiste(id_avis, artiste)
avis_tag(id_avis, nom_tag)

Clés étrangères : **user_id**, **utilisateurs_2** dans amitie **artiste**, **lieu** dans concerts_a_venir
artiste, **lieu** dans concerts_finis **user_id**, **artiste** dans interet **user_id**, **artiste** dans
photos_videos **artiste** dans morceaux **user_id** dans playlist **user_morceau**, **user_playlist**
dans morceau_de_playlist **user_id** dans avis **user_morceau** dans avis_morceau
user_playlist dans avis_playlist **artiste** dans avis_concert_fini **lieu** dans avis_lieu **artiste**
dans avis_artiste font toutes references **user_id** dans utilisateurs

sous_genre_de dans tags **nom_tag** dans avis_tag font reference a **nom** dans tags

concert_date dans interet fait reference a **concert_date** dans concerts_a_venir

concert_date dans photos_videos **concert_date** dans avis_concert_data font reference a
concert_date dans concert_fini

nom_morceau dans morceau_de_playlist **nom_morceau** de avis_morceau font reference a
nom_morceau dans morceaux

nom_playlist dans morceau_de_playlist **nom_playlist** dans avis_playlist font reference a
nom dans playlists

id_avis dans avis_morceau, avis_playlist, avis_concert_fini, avis_avis_lieu, avis_artiste,
avis_tag font reference a **id_avis** dans avis

III. Les requêtes implémentées

- Les concerts ayant lieu dans les n jours à venir
- Les concerts pris en photo par un utilisateur
 - Sous-requête dans le WHERE
- Nombre de concerts à venir pour chaque artiste
- Nombre d'amis d'un utilisateur ayant au moins un follower
- Les utilisateurs qui n'ont pas d'amis
 - Condition de totalité
 - Une requête avec sous-requête corrélée
 - Une requête avec agrégation (LEFT JOIN)
- La moyenne des notes de chaque concert
 - Sous-requête dans le FROM
 - Porte sur trois tables
- Avoir les amis d'amis d'un utilisateur
 - Requête récursive
- Trouver le nombre d'avis postés par un utilisateur (supérieur à 2)
 - GROUP BY - HAVING
- Trouver les morceaux dans une playlist
- Trouver la moyenne des avis de chaque utilisateur (supérieur à 3)
 - GROUP BY - HAVING
- Trouver les artistes ayant fini des concerts ayant plus de 1500 participant
- Trouver le nombre de sous-genre d'un genre
 - Auto-jointure + LEFT JOIN
 - Deux requêtes donnant des résultats différents à cause des NULL.
- Classe les utilisateurs en fonction de leur nombre d'avis
 - Fenêtrage
- La moyenne des moyennes des notes des lieux
 - Calcul de deux agrégats
- Classe les artistes par leur nombre d'avis
 - Fenêtrage
- Classe les artistes en fonctions de leur note moyenne
 - Fenêtrage

IV. Indice de recommandation

Pour chaque avis laissé par un utilisateur donné, on calcule la moyenne des notes en fonction des tags qu'il a associé à ces avis. On garde les moyennes supérieures ou égales à 3. Ensuite, on cherche les artistes ayant produit des morceaux comportant les tags renvoyés par la requête précédente et on recommande les morceaux de ces artistes.