



# Rendu 2 de Projet en Bases de Données

## BIN5\_WAVE

## Schéma Relationnel

artiste ( id\_artiste, nom, prenom, nationalite, date\_naiss, date\_mort )

morceau ( id\_morceau, titre, donnee, paroles, genre, id\_groupe )

groupe ( id\_groupe, nom, date\_creation, nationalite, genre )

utilisateur ( id\_user, pseudo, date\_inscription, email, mdp )

playlist ( id\_playlist, titre, visible, description, id\_user )

album ( id\_album, titre, date\_parution, image, descript, auteur )

participe ( id\_artiste, id\_morceau )

appartient ( id\_artiste, id\_groupe, arrivée, depart )

suis\_groupe ( id\_groupe, id\_user, parametre )

ecoute ( id\_morceau, id\_user, date\_ecoute )

dans\_playlist ( id\_morceau, id\_playlist, numero )

suis\_user ( id\_user, id\_user2 )

dans\_album ( id\_morceau, id\_album, numero )

FK:

id\_artiste dans (appartient,participe) fait ref à la clé primaire id\_artiste de artiste

id\_groupe / id\_groupe dans (appartient, suis\_groupe, morceau) fait ref à la clé primaire id\_groupe de groupe

id\_user / id\_user dans (suis\_groupe, suis\_user, playlist, ecoute) fait ref à la clé primaire id\_user de utilisateur

id\_user2 dans (suis\_user) fait ref à la clé primaire id\_user de utilisateur

id\_album dans (dans\_album) fait ref à la clé primaire id\_album de album

id\_morceau dans (participe, ecoute, dans\_album, dans\_playlist) fait ref à la clé primaire id\_morceau de morceau

auteur dans (album) fait ref à la clé primaire id\_groupe dans groupe.

Légende : Clé primaire, Clé étrangère

## Contraintes usuelles

artiste ( id\_artiste, nom, prenom, nationalite, date\_naiss, date\_mort )  
morceau ( id\_morceau, titre, donnee, paroles, genre, id\_groupe )  
groupe ( id\_groupe, nom, date\_creation, nationalite, genre )  
utilisateur ( id\_user, pseudo, date\_inscription, email, mdp )  
playlist ( id\_playlist, titre, visible, description, id\_user )  
album ( id\_album, titre, date\_parution, image, descript, auteur )  
participe ( id\_artiste, id\_morceau )  
appartient ( id\_artiste, id\_groupe, arrivee, depart )  
suis\_groupe ( id\_groupe, id\_user, parametre )  
ecoute ( id\_morceau, id\_user, date\_ecoute )  
dans\_playlist ( id\_morceau, id\_playlist, numero )  
suis\_user ( id\_user, id\_user2 )  
dans\_album ( id\_morceau, id\_album, numero )

Légende :

- ON DELETE : SET NULL / **CASCADE**
- **NOT NULL**
- **UNIQUE**

Remarque : Les NOT NULL ne reprennent pas les clés primaires qui sont forcément définies, les NOT NULL dessus sont redondants.

## Contraintes spécifiques

chronologie\_appartient : La date de départ est soit indéfinie ou après celle d'arrivée

suis\_groupe\_parametre : L'attribut parametre doit être égal soit à 'A' pour Album seulement, 'M' pour suivre les morceaux seulement ou 'AM' pour les deux.

email\_format : l'attribut email doit être de la forme ...@...

user\_mdp\_length : la taille du mot de passe doit être supérieur à 6

## Type de données

serial : tous les identifiants d'entités/table sans clé primaire à la fois étrangère. (Ce sont implicitement des ints)

int : numero(dans\_album), numero(dans\_playlist), les FK faisant référence à des serials.

varchar(50) : nom(artiste), prenom(artiste), nom(groupe), titre(morceau), email(utilisateur), titre(album)

varchar(30) : genre(groupe), genre(morceau), mdp(utilisateur), titre(playlist)

varchar(20) : pseudo(utilisateur)

char(3): nationalite(artiste), nationalite(groupe) (sous format ISO 3166-1 alpha-3)

date : date\_naiss(artiste), date\_mort(artiste), date\_creation(groupe), date\_parrution(album), arrivee(appartient), depart(appartient)

timestamp : date\_ecoute(ecoute)

## Liste d'exemple de cas de figures représentés par les données factices de la base de donnée

- Artistes qui ont une carrière solo, représenté dans la table groupe avec leurs noms d'artistes
- Deux artistes formant un groupe, qui est désormais dissous mais leurs carrières solo respectives sont toujours en cours
- Un artiste possédant 2 albums
- Un artiste possédant 1 album
- Un artiste ne possédant aucun album
- Un album contenant plusieurs morceaux
- Un album contenant un unique morceau
- Album vide
- Un utilisateur possédant 1 playlist en public
- Un utilisateur possédant 2 playlists dont une privé et une en publique
- Un utilisateur possédant une playlist en privé
- Un utilisateur possédant une playlist vide
- Un artiste qui a composé plusieurs morceaux
- Un artiste qui a composé un seul morceau
- Un groupe qui a composé un morceau
- Un groupe qui a composé un morceau en feat avec un artiste extérieur
- Utilisateur qui suit un groupe avec le paramètre A
- Utilisateur qui suit un groupe avec le paramètre M
- Utilisateur qui suit un groupe avec le paramètre AM
- Un utilisateur qui suit 2 autres utilisateurs
- Un utilisateur qui suit 1 utilisateur
- Une playlist contenant différent morceaux de différent artistes
- Une playlist contenant 1 seul morceau
- Un utilisateur écoutant plusieurs fois le même morceau à des instant différent
- Deux utilisateur écoutant le même morceaux au même moment
- Un utilisateur écoutant deux musiques au meme moment (shuffling rapide)
- Un utilisateur ayant écouté deux morceaux différents

## Choix personnels

- Suppression de la ternaire reliant un album, un morceau et un groupe, au profit de deux binaires suite aux retours du rendu 1.
- Ajout de l'attribut paramètre à suis\_groupe pour spécifier le type de suivi suite aux retours du rendu 1.
- Choix d'utiliser ON DELETE SET NULL sur id\_groupe dans morceau et sur auteur dans album car les lignes peuvent garder un sens lors que les lignes référencées sont supprimées.
- Choix d'utiliser ON DELETE CASCADE sur toutes les autres FK car ils perdent leur sens quand les lignes référencées sont supprimées.
- Unicité du pseudo et de l'email d'inscription pour les utilisateurs.
- L'intégralité des contraintes spécifiques.