Université Gustave Eiffel

L2 – Bases de données

Projet The Wave

Etape 2

Clément GAUDET / Maxime JAILLARD
26/11/20

Sommaire

TABLE DES MATIÈRES

Schéma relationnel

artiste(idA, nomA, prenom, nationA, dateNais, dateMort)
membre(idMe, role, dateDeb, dateFin, #idA, #idG)
groupe(idG, nomG, dateCrea, nationG, genre)
morceau(idMo, titreM, duree, paroles, audio, #idG)
album(idAl, titreA, dateParu, couv, descA, #idG)
utilisateur(pseudo, email, dateInsc, mdp)
playlist(idP, titre, descP, privee, #pseudo)
participe(#idA, #idMo)
albumContient(#idAl, #idMo, num)
playlistContient(#idP, #idMo, num)
suitGroupe(#pseudo, #idG)
suitUtilisateur(#suit, #suivi)
historique(#pseudo, #idMo, dateHeure)

Clés étrangères	Colonne référencée
• membre.idA • participe.idA	artiste.idA
• membre.idG • morceau.idG • album.idG • suitGroupe.idG	groupe.idG
• albumContient.idAl	album.idAl
• playlistContient.idP	playlist.idP
• playlistContient.idMo • historique.idMo • albumContient.idMo • participe.idMo	morceau.idMo
• suitGroupe.pseudo • playlist.pseudo • suitUtilisateur.suit • suitUtilisateur.suivi • historique.pseudo	utilisateur.pseudo

Une **correction** par rapport au schéma EA a été apportée ici. En effet l'attribut de nationalité n'était pas dans l'entité artiste. Elle a été ajoutée dans le schéma relationnel,

Implémentation sous postgreSQL

Généralités

- Toutes les colonnes clé primaire en « id*xx* » sont de type *serial*, permettant de laisser le SGBD gérer automatiquement la numérotation
- La plupart des colonnes demandant une chaine de caractères de taille modeste sont de type *varchar(50)*
- Sauf indiqué, la plupart des colonnes ont une contrainte NOT NULL

Table artiste

- *dateNais, dateMort* et *nationA* peuvent être NULL, puisqu'évidemment, beaucoup d'artistes ne sont pas encore morts et certains n'ont pas renseigné d'informations de naissance ou de nationalité.
- Une contrainte UNIQUE sur le tuple (*nomA, prenom*) existe pour éviter que le même artiste apparaisse plusieurs fois dans la table
- Une contrainte CHECK sur *dateMort* permet de s'assurer qu'elle soit postérieure à la date de naissance

Table groupe

- *nationG* peut être NULL, supposant qu'un groupe pourrait ne pas avoir une nationalité particulière

Table membre

- *dateFin* peut être NULL, puisqu'un artiste peut encore à l'heure actuelle occuper un rôle dans le groupe en question
- Une contrainte CHECK permet de s'assurer que *dateFin* est postérieure à *dateDeb*

Table album

- *couv* est de type *varchar(200)*, cette colonne est sensée contenir le nom du fichier contenant l'image (l'URL sera complétée au niveau applicatif, de même pour les autres colonnes fonctionnant de la même manière)
- *descA* est de type *text*, car pouvant potentiellement être longue
- *couv* et *descA* peuvent être NULL, pouvant supposer que les deux peuvent manquer

Table utilisateur

- Contrainte UNIQUE sur email car on suppose, comme c'est le cas pour la grande majorité des sites sur internet, que chaque email ne peut être utilisé que pour un compte

Table playlist

- *descP* est de type *text* car une description peut faire plusieurs paragraphes.
- *privee* est de type *boolean* car il n'existe que deux états pour le caractère privé d'une playlist (oui/non)

Tables participe, suitGroupe

- Ces tables sont uniquement composées de tuples de clés étrangères pris comme clé primaire de leur table.

Table suitUtilisateur

- Un utilisateur ne peut pas se suivre lui-même, ainsi une contrainte CHECK s'assure que celui qui suit et celui qui est suivi ne sont pas les mêmes.

Table albumContient

- *num* est de type *int* car il s'agit d'un simple indice de position.
- Contrainte CHECK (*num* > 0) pour empêcher les entrées où *num* est nul ou négatif.
- Contrainte UNIQUE sur le tuple (*idAl*, *num*) pour assurer de ne pas avoir de doublon de numéros dans un même album

Table playlistContient

- Contrairement à *albumContient*, ici la clé primaire est (*idP, num*), et aucune contrainte d'unicité n'existe sur *idMo*, permettant à un utilisateur de mettre plusieurs fois le même morceau dans une même playlist s'il le souhaite.
- *num* est de type *int* car il s'agit d'un simple indice de position.
- Contrainte CHECK (*num* > 0) pour empêcher les entrées où *num* est nul ou négatif.

Table historique

- *dateHeure* est de type *timestamp* pour conserver l'heure et la date d'écoute
- (*pseudo*, *dateHeure*) est clé primaire au lieu de (*pseudo*, *idMo*) car le but est d'empêcher deux entrées à horodatages identiques. En revanche, un utilisateur peut très bien écouter plus d'une fois le même morceau.

Contraintes non assurées

- L'implémentation **ne permet pas en elle-même d'assurer une numération correcte des albums et playlists**. Elle permet d'éviter des doublons, mais pas que la numération soit continue. Il peut en effet y avoir des trous.
- **La date de début du rôle d'un artiste peut être antérieure à la création du groupe.** Cette incohérence n'est pas interdite dans la base en l'état. Il en va de même pour les rapports incohérents entre colonnes de date n'appartenant pas à la même table (le rôle d'un artiste débute avant sa naissance, date dans l'historique avant la création du compte…).