

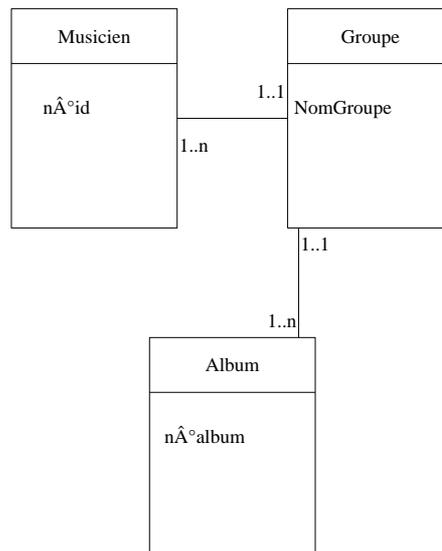
DS d'informatique n°2 : corrigé

PTSI Lycée Eiffel

17 mars 2017

Exercice 1

1. Non, puisque Instrument est un attribut de l'entité Musicien, on ne peut avoir qu'un seul instrument par musicien. Pour changer cela, on pourrait au choix considérer Instrument comme une association entre les musiciens et les groupes (un musicien pourrait alors jouer de différents instruments dans divers groupes, mais ne pourrait toujours jouer que d'un seul instrument au sein d'un groupe donné) ou, encore mieux, créer une entité séparée Instrument avec une association de cardinalité 1..n 1..n entre Musicien et Instrument.
2. On va bien sûr créer trois entités correspondant à ce que précise l'énoncé, avec des associations entre Musicien et Groupe, et entre Groupe et Album (puisque un album ne fait intervenir qu'un seul groupe). On a indiqué uniquement le premier attribut de chaque entité pour alléger le schéma :



3. L'association entre Groupe et Album étant de cardinalité 1..1 1..n, on se contentera d'ajouter un attribut N°Groupe dans la relation Album. Par contre, l'association plus complexe entre Musicien et Groupe va nécessiter la création d'une nouvelle relation qu'on appellera Zoinx (ou tout autre nom plus approprié). Les clés primaires sont indiquées en gras, les clés secondaires en italique dans le schéma suivant :
- Musicien (**N°Mus** INTEGER, *Nom* VARCHAR(20), *Prénom* VARCHAR(20), *Instrument* VARCHAR(20))
 - Groupe (**N°Groupe** INTEGER, *NomGroupe* VARCHAR(30), *Genre* VARCHAR(20), *Nbremusiciens* INTEGER)
 - Zoinx (**N°Mus** INTEGER, **N°Groupe** INTEGER)

- Album (**N°Album** INTEGER, Titre VARCHAR(30), *N°Groupe* INTEGER, Durée INTEGER)
4. (a) $\Pi_{\text{Nom, Prénom}}(\text{Musicien})$
 (b) $\Pi_{\text{Titre}}(\sigma_{\text{genre}=\text{metal symphonique}}(\text{Groupe} \bowtie_{\text{Groupe.N°Groupe}=\text{Album.N°Groupe}} \text{Album}))$
 5. ALTER TABLE Musiciens ADD Age INTEGER
 DELETE FROM Musiciens WHERE Instrument=Accordéon
 6. (a) SELECT NomGroupe FROM Groupe WHERE Nbremusiciens >4
 (b) SELECT Nom, Prénom FROM Musiciens WHERE Nom LIKE 'Z%' AND Prénom LIKE 'Z%'
 (c) SELECT Nom, Prénom FROM Musiciens M, Groupe G, Album A, Zoinx Z WHERE Z.N°Mus=M.N°Mus AND Z.N°Groupe=G.N°Groupe AND A.N°Groupe=G.N°Groupe AND A.Titre='Droit de véto'
 (d) SELECT NomGroupe FROM Groupe, Album WHERE Album.N°Groupe=Groupe.N°Groupe AND Durée<40 AND Genre='rap'

Remarque : pour la dernière recherche, un même groupe sera affiché plusieurs fois s'il a fait plusieurs albums durant moins de 40 minutes. Il existe des commandes en SQL pour n'afficher qu'une seule fois chaque résultat possible mais nous ne les avons pas vues en cours.

Exercice 2

1. Tout simplement parce qu'une même langue peut être parlée dans plusieurs pays (et plusieurs langues parlées dans un même pays), c'est la façon la plus simple d'identifier les instances de cette relation.
2. L'attribut Codepays de la relation Villes est une clé secondaire, et on peut considérer que l'attribut Capitale de la relation Pays est aussi un attribut secondaire (si la base est bien construite, ce qui devrait apparaître dans cet attribut devrait d'ailleurs être le Codeville de la ville correspondante et pas son Nom).
3. (a) SELECT Nom, Population FROM Pays WHERE Continent='Europe'
 (b) SELECT NomLangue, Pourcentage FROM Langues,Pays WHERE Pays.Codepays=Langues.Codepays AND Nom='Chine'
 (c) SELECT Villes.Nom FROM Pays,Villes,Langue WHERE Capitale=Villes.Nom AND Langues.Codepays=Pays.Codepays AND Nomlangue='Français'
 (d) SELECT AVG(Esperancedevie) FROM Pays WHERE Continent='Afrique'
 Remarque : cette commande va faire la moyenne de toutes les espérances de vie des pays africains, et en fait pas vraiment calculer l'espérance de vie moyenne en Afrique (il faudrait pondérer chaque pays par sa population, ce qui est plus compliqué).
 (e) SELECT COUNT(Nom) FROM Villes WHERE Population > 1000000
 (f) SELECT SUM(Population*Pourcentage) FROM Pays, Langues WHERE Pays.Codepays=Langues.CodePays AND Nomlangue='Français'