

OBJECTIF : Apprentissage des commandes SQL du LDD et du LMD

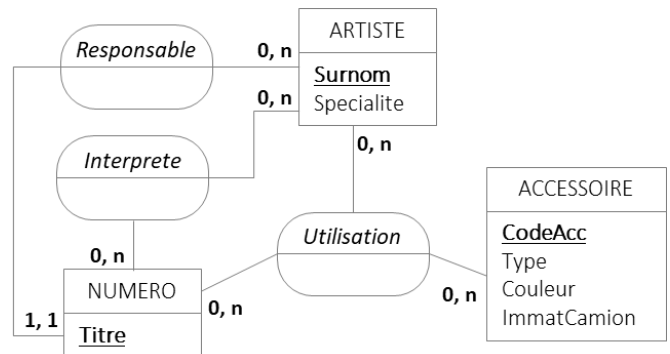
CONTEXTE DE L'ÉTUDE : BD Cirque présentée dans le TD5

RAPPEL DES RÈGLES DE GESTION ET DU SEA, QUI ONT CONDUIT AU SCHÉMA RELATIONNEL DE LA BD CIRQUE

RÈGLES DE GESTION

- RG1.** Un artiste du cirque est identifié par un surnom unique. On enregistre sa spécialité principale (*jongleur, clown, acrobate ou équilibriste*)
- RG2.** Un numéro du cirque est interprété par des artistes du cirque. Il est identifié par son titre et a un responsable qui fait partie des artistes du cirque.
Le responsable d'un numéro n'interprète pas forcément ce numéro.
- RG3.** Les interprètes d'un numéro peuvent utiliser des accessoires.
Un accessoire est identifié par un code unique.
Ses autres caractéristiques sont : son type (*ballon, cerceau, foulard, quilles, trapèze, corde, ou autre*) et le numéro d'immatriculation (*conforme SIV¹*) du camion dans lequel il est stocké quand il n'est pas utilisé.

SCHÉMA ENTITÉS-ASSOCIATIONS (SEA)



C.I. : Tout couple (SurNom, Titre) faisant partie des occurrences de *Utilisation* doit faire partie des occurrences de *Interprete*

CONNEXION AU SGBD POSTGRESQL

À PARTIR DE CE TP, VOUS TRAVAILLerez SUR VOTRE PROPRE BASE

- votre **base** de données a le même nom que **votre login Unix**
- votre **utilisateur** PostgreSQL a le même nom que **votre login Unix**
- votre **mot de passe** est **votre login Unix**

CONNEXION À PostgreSQL : `psql -h postgres-info login -U login`
puis saisie de votre login Unix

DÉCONNEXION : `\q`

PRÉLIMINAIRES

- Dans **votre** répertoire R1.05, **créez** un répertoire **TP3_Cirque**
- Ouvrez un terminal et placez-vous dans le répertoire **TP3_Cirque**, puis **effectuez la commande** :
`cp -r /users/info/pub/1a/R1.05/TP3_1/*.* .`

VÉRIFIEZ QUE VOTRE RÉPERTOIRE **TP3_Cirque** CONTIENT LES FICHIERS **insert.sql**, **test.sql** et **new_data.sql**

- Dans votre répertoire **TP3_Cirque**, **créez** :
 - * un fichier **create.sql** où vous écrirez les instructions de *création des relations* de la BD Cirque
 - * un fichier **drop.sql** où vous écrirez les instructions de *suppression des relations* de la BD Cirque

¹ <https://www.cartegrise.com/carte-grise-detail/numero-immatriculation>

Partie A : Création / Suppression des relations de la BD cirque

SCHÉMA RELATIONNEL ET TYPE DES ATTRIBUTS cf. TD5

ARTISTE (Surnom, Specialite)

Attribut	Type	
Surnom	varchar(20)	surnom qui identifie un artiste du cirque
Specialite	varchar	la spécialité principale de cet artiste

NUMERO (Titre, #Responsable)

Attribut	Type	
Titre	varchar	intitulé qui identifie un numéro
Responsable	varchar(20)	surnom du responsable de ce numéro – référence la clé primaire de ARTISTE

INTERPRETE(#Surnom, #Titre)

Attribut	Type	
Surnom	varchar(20)	surnom d'un artiste – référence la clé primaire de ARTISTE
Titre	varchar	intitulé d'un numéro interprété par cet artiste – référence la clé primaire de NUMERO

ACCESSOIRE(CodeAcc, Type, Couleur, ImmatCamion)

Attribut	Type	
CodeAcc	numeric(3)	code qui identifie un accessoire
Type	varchar(20)	type de cet accessoire (exemple : ballon)
Couleur	varchar	couleur de cet accessoire
ImmatCamion	char(9)	immatriculation du camion où est rangé cet accessoire quand il n'est pas utilisé

UTILISATION(#CodeAcc, #(SurNom, Titre))

Attribut	Type	
CodeAcc	numeric(3)	code d'un accessoire
Surnom	varchar(20)	surnom d'un artiste qui utilise cet accessoire
Titre	varchar	titre d'un numéro dans lequel cet artiste utilise cet accessoire
(SurNom, Titre) référence la clé primaire de INTERPRETE		

A1. ÉBAUCHE D'IMPLÉMENTATION DES RELATIONS DE LA BD CIRQUE

- Dans votre fichier **create.sql** écrivez les instructions de **création** des relations de la BD en vous limitant pour chacune d'entre-elles, à :
 - ✓ la définition du nom et du type des attributs
 - ✓ l'expression des contraintes de clé primaire et, s'il y lieu, de clé étrangère
- Enregistrez le fichier **create.sql**, ouvrez un terminal à partir de votre répertoire **TP3_Cirque**
- Connectez-vous à PostgreSQL
- Exécutez le fichier **create.sql** en tapant la commande **\i create.sql**
- Affichez les informations relatives aux relations créées à l'aide de la commande **\d** suivie du nom de la relation à vérifier (exemple **\d ARTISTE**)

A2. INSTRUCTIONS DE SUPPRESSION DES RELATIONS DE LA BD CIRQUE

- Dans votre fichier **drop.sql** écrivez les instructions de **suppression** des relations de la BD en respectant la contrainte énoncée ci-dessous :

CONTRAINTES : Utilisation de **CASCADE** "interdite"...

Vous devez réfléchir à l'ordre dans lequel les relations doivent être supprimées

- Enregistrez le fichier **drop.sql**
- Exécutez-le en tapant la commande **\i drop.sql**

NOTE : Après l'exécution du fichier **drop.sql**, aucune relation ne doit exister dans votre base.
 Vous pouvez le vérifier en tapant la commande **\d**
 Si ce n'est pas le cas, corrigez votre fichier **drop.sql** et réexécutez-le

A3. AJOUT DE CONTRAINTES D'ATTRIBUT DANS LES RELATIONS DE LA BD CIRQUE

- Dans votre fichier **create.sql** ajoutez les contraintes suivantes :
 - ✓ l'attribut **Specialite** doit être renseigné et prendre ses valeurs parmi *jongleur, clown, acrobate, équilibriste*
ATTENTION : cette contrainte devra être nommée explicitement (*nom de la contrainte* : **c_specialite**)
 - ✓ l'attribut **Responsable** doit être renseigné
 - ✓ l'attribut **ImmatCamion** doit respecter le format 'XX-XXX-XX' (X représentant un caractère quelconque) et doit être renseigné
 - ✓ la valeur de l'attribut **Type** doit faire partie de *ballon, cerceau, foulard, quilles, trapèze, corde, autre* avec *autre* comme valeur par défaut
 - ✓ la valeur de l'attribut **Couleur** doit faire partie de *blanc, noir, rouge, bleu, vert, jaune, orange, violet* ; cet attribut peut ne pas être renseigné
- Enregistrez le fichier **create.sql** et réexécutez-le en tapant la commande **\i create.sql**
En cas d'erreur signalée par le SGBD
 1. Supprimez les relations de la BD en tapant la commande **\i drop.sql**
 2. Corrigez l'erreur signalée par le SGBD lors de l'exécution du fichier **create.sql**
 3. Réexécutez le fichier **create.sql**

A4. TESTS DE CONFORMITÉ DES RELATIONS CRÉÉES

Le fichier **test.sql** contient des instructions à exécuter successivement pour vérifier si l'ensemble des contraintes ont bien été exprimées lors de la création des relations de la BD.

- Ouvrez le fichier **test.sql** puis, pour chaque instruction respectez la procédure détaillée ci-dessous

TEST DE L'INSTRUCTION n°X (les instructions précédentes ayant donné le résultat attendu...)	
1 ^{ère} étape :	Copiez le code de l'instruction n°X (ctrl C) Collez-là (ctrl + Shift + V) <u>dans le terminal où vous êtes connectés à PostgreSQL</u>
2 ^{ème} étape :	Vérifiez si son résultat est conforme à celui attendu
3 ^{ème} étape :	<i>Si vous n'obtenez pas le résultat attendu ...</i> <ul style="list-style-type: none">- Dans le fichier create.sql <u>corrigez la (ou les) ligne(s) susceptible(s) d'en être la cause</u>- Exécutez successivement les fichiers drop.sql et create.sql- Réexécutez les instructions du fichier test.sql <u>jusqu'à l'instruction n°X</u>- Tentez à nouveau l'exécution de l'instruction n°X <i>Si vous obtenez le résultat attendu, procédez au test de l'instruction suivante, s'il y en a une ...</i>

- Dès que l'exécution de toutes les instructions du fichier **test.sql** donne le résultat attendu, supprimez les relations de la base, puis recréez-les

Partie B : Peuplement des relations de la BD cirque

B1. PEUPLEMENT INITIAL DES RELATIONS DE LA BD CIRQUE

Le fichier **insert.sql** contient les instructions de peuplement initial des relations de la BD.

- Exécutez le fichier **insert.sql** - **AUCUNE ERREUR NE DOIT SE PRODUIRE !!!**
- Affichez le contenu de chaque relation

B2. INSERTION DE NOUVELLES DONNÉES DANS LES RELATIONS DE LA BD CIRQUE

L'objectif de cette partie est de travailler les deux formes de la commande INSERT en répondant à de nouveaux besoins en termes de données de la BD.

Le fichier **new_data.sql** sera complété par vos soins pour répondre à ces nouveaux besoins.

-
- BESOIN N°1. Embauche d'un *jongleur*, d'un *clown*, d'un *acrobate* et d'un *équilibriste*
- BESOIN N°2. Création d'un nouveau numéro sous la responsabilité de *Etoile*
- BESOIN N°3. Participation de *Marie* et de *Volante* à ce nouveau numéro
- BESOIN N°4. Acquisition d'un *trapèze* de code *21*, de couleur non renseignée et qui sera rangé dans le camion où est rangé l'accessoire de code *18*
- BESOIN N°5. *Bozzo* aimerait utiliser des cerceaux dans le numéro intitulé *Les Zappatas* : faire en sorte qu'il puisse utiliser tous les cerceaux, sachant qu'il n'en utilisait aucun dans ce numéro
- BESOIN N°6. Participation de tous les artistes qui n'y participaient pas déjà, au numéro *Les Zappatas*
- BESOIN N°7. Création d'un numéro pour fêter la nouvelle année, avec *Bozzo* comme responsable
- BESOIN N°8. Participation de tous les artistes à ce nouveau numéro
- BESOIN N°9. Pour ce nouveau numéro, les artistes utiliseront les accessoires qu'ils utilisent dans les autres numéros qu'ils interprètent. Faire le nécessaire pour que cela soit pris en compte...
-

- Ouvrez le fichier **new_data.sql**
- Pour chaque besoin à traiter, respectez la procédure détaillée ci-dessous :

TRAITEMENT DU BESOIN n°X

1^{ère} étape : Dans le fichier **new_data.sql**

- Écrivez la (ou les) instruction(s) répondant au besoin n°X
- Enregistrez le fichier **new_data.sql**

2^{ème} étape : Copiez le code produit, puis collez-le dans le terminal où vous êtes connectés à PostgreSQL

► Dans les cas où le code produit n'a conduit à aucune insertion :

- * cas 1 – le SGBD a signalé une erreur
- * cas 2 – **INSERT 0 0** a été affiché par le SGBD

Reprenez la 1^{ère} étape pour tenter de corriger votre code...

► Dans le cas où des lignes ont été insérées :

- Affichez la relation où les données ont été insérées
- Vérifiez que les données insérées répondent bien au besoin n°X
- * Si c'est le cas, passez au besoin suivant, s'il y en a un...

* Si ce n'est pas le cas...

- mettez en commentaire votre réponse au besoin n°X, puis enregistrez le fichier **new_data.sql**
- exécutez successivement les fichiers **drop.sql**, **create.sql** et **new_data.sql**

Reprenez la 1^{ère} étape pour tenter de corriger votre code...