

## TDM4– SQL - CREATION DE TABLES AVEC CONTRAINTES

### Base de données Musiques

Afin de faciliter l'écriture de certaines requêtes, voici des liens utiles vers la documentation PostgreSQL (le PDF contient des liens web) : création de table, création d'indices, modification de la structure des tables, ajout de données, suppression de données.

Vous trouverez à la fin de cet énoncé les requêtes SQL de création de tables pour une base de données nommée Musiques.

Pour les questions ci-dessous, vous manipulerez votre base sur le serveur webtp par l'intermédiaire d'une interface web appelée Phppgadmin. Dans un navigateur, ouvrez le lien <http://webtp.fil.univ-lille1.fr/phppgadmin/> puis cliquez à gauche sur « postgresQL » pour vous connecter. Phppgadmin est un client du SGBD qui permet de visualiser les tables et leurs contenus et de réaliser de nombreuses opérations de façon « graphique ». Il permet aussi d'exécuter les requêtes SQL de son choix.

**Q1.** En vous basant sur les requêtes du fichier de création de tables, retrouver le MLD puis le MCD de cette base.

Un groupe est représenté par un enregistrement de la table artiste dont le pseudo est le nom du groupe et les nom et prénom sont vides.

**Q2.** Quels sont les résultats des requêtes suivantes ? Expliquez l'erreur si vous pensez qu'il y en a une.

```
INSERT INTO Artiste(pseudo) VALUES ('Beatles');
```

```
INSERT INTO Artiste(pseudo) VALUES ('beatles');
```

**Q3.** Quelle(s) requête(s) permette(nt) d'ajouter les membres des Beatles dans la table Artistes ? Voici leurs prénom et nom : John Lennon, Paul McCartney, Georges Harisson, Ringo Starr

**Q4.** Quelle requête permet d'obtenir le numArtiste du groupe Beatles ?

**Q5.** Quelle table permet d'associer les membres des Beatles à leur groupe ?

Lors d'une insertion, il est possible d'obtenir les valeurs à remplir en interrogeant d'autres tables à l'aide de requêtes mises entre parenthèses. Par exemple :

```
INSERT INTO tableX(col) VALUES ((SELECT colY FROM tableY WHERE condition));
```

**Q6** Utilisez cette syntaxe pour écrire des requêtes qui associent les membres des Beatles à leur groupe, en allant chercher leur numArtiste automatiquement.

**Q7.** En vous inspirant de la requête présente dans le fichier de création des tables, bloquez la possibilité qu'une chanson apparaisse deux fois dans la base (même titre quelque-soit la casse et même durée).

**Q8.** Est-ce que les requêtes suivantes produisent une erreur suite à l'ajout de la contrainte ?

```
INSERT INTO Chanson(titre, duree) VALUES
```

```
('Yellow Submarine', '2:39'),
```

```
('She loves you', '2:19'),
```

```
('Imagine', '3:1'),
```

```
('She loves you', '2:50');
```

```
INSERT INTO Chanson(titre, duree) VALUES ('imagine', '03:01');
```

**Q9.** En tenant compte de l'état de la base suite à l'exécution des requêtes précédentes, quel est le résultat de la requête suivante ? Expliquez l'erreur si vous pensez qu'il y en a une et corrigez la requête pour qu'elle fonctionne.

```
INSERT INTO Contribuer_a VALUES
```

```
((SELECT numArtiste FROM Artiste WHERE pseudo='Beatles'),  
 (SELECT numChanson FROM Chanson WHERE titre='She loves you'),  
 'Interprète');
```

**Q10.** Pourquoi la structure actuelle de la base ne permet pas qu'un artiste ait plusieurs rôles pour une chanson ? Que faudrait-il faire pour que cela soit le cas ?

**Q11.** Quels sont les résultats des requêtes suivantes ? Expliquez l'erreur si vous pensez qu'il y en a une et corrigez la requête pour qu'elle fonctionne.

```
INSERT INTO Album(titre, date_sortie) VALUES ('Revolver', '1966-08-05');
```

```
INSERT INTO Album(titre, date_sortie) VALUES ('Imagine', '9/9/1971');
```

```
INSERT INTO Album(titre) VALUES('Anthology 1');
```

**Q12.** Quels sont les résultats des requêtes suivantes ? Expliquez l'erreur si vous pensez qu'il y en a une et expliquer ce qu'il faut faire pour qu'elle fonctionne.

-- 1.

```
INSERT INTO Se_composer_de VALUES
```

```
((SELECT numAlbum FROM Album WHERE titre='revolver'),  
 (SELECT numChanson FROM Chanson WHERE titre = 'Yellow Submarine'));
```

-- 2.

```
INSERT INTO Se_composer_de VALUES
```

```
((SELECT numAlbum FROM Album WHERE titre='Imagine'),  
 (SELECT numChanson FROM Chanson WHERE titre = 'Imagine' ));
```

-- 3.

```
INSERT INTO Se_composer_de VALUES
```

```
((SELECT numAlbum FROM Album WHERE titre='Anthology 1'),  
 (SELECT numChanson FROM Chanson WHERE titre = 'She loves you' and duree='2:50'));
```

**Q13.** Quel est le résultat de la requête suivante ? Expliquez l'erreur si vous pensez qu'il y en a une et expliquer ce qu'il faut faire pour qu'elle fonctionne.

```
DELETE FROM Chanson WHERE duree='3:1';
```

## Requêtes de création de la base

*-- Tables sans clé étrangère*

```
CREATE TABLE Album (  
    numAlbum SERIAL PRIMARY KEY,  
    titre VARCHAR(200) NOT NULL,  
    date_sortie DATE NOT NULL  
);
```

```
CREATE TABLE Artiste (  
    numArtiste SERIAL PRIMARY KEY,  
    nom VARCHAR(200),  
    prenom VARCHAR(200),  
    pseudo VARCHAR(200) -- Pseudo de l'artiste ou nom du groupe  
);
```

*-- Pour s'assurer que 2 artistes/groupes n'aient pas le même pseudo,  
-- quelque-soit la casse, il est possible de passer par la création d'un index*

```
CREATE UNIQUE INDEX Artiste_pseudo ON Artiste (lower(pseudo));
```

```
CREATE TABLE Chanson (  
    numChanson SERIAL PRIMARY KEY,  
    titre VARCHAR(500) NOT NULL,  
    duree INTERVAL NOT NULL  
);
```

*-- Tables avec clés étrangères*

*-- Remarque : certaines contraintes sont nommées pour pouvoir les modifier*

*-- Remarque : deux contraintes de tables différentes peuvent avoir le même nom*

```
CREATE TABLE Se_composer_de (  
    numAlbum INTEGER,  
    numChanson INTEGER,  
    PRIMARY KEY (numAlbum, numChanson),  
    CONSTRAINT fk_Album  
        FOREIGN KEY (numAlbum) REFERENCES Album,  
    CONSTRAINT fk_Chanson  
        FOREIGN KEY (numChanson) REFERENCES Chanson  
);
```

```
CREATE TABLE Etre_membre_de (  
    numArtiste INTEGER,  
    numGroupe INTEGER,  
    PRIMARY KEY (numArtiste, numGroupe),  
    CONSTRAINT fk_Artiste  
        FOREIGN KEY (numArtiste) REFERENCES Artiste(numArtiste),  
    CONSTRAINT fk_Groupe  
        FOREIGN KEY (numArtiste) REFERENCES Artiste(numArtiste)  
);
```

```
CREATE TABLE Contribuer_a (  
    numArtiste INTEGER,  
    numChanson INTEGER,  
    role VARCHAR(100),  
    PRIMARY KEY (numArtiste, numChanson),  
    CONSTRAINT fk_Artiste  
        FOREIGN KEY (numArtiste) REFERENCES Artiste  
        ON DELETE CASCADE,  
    CONSTRAINT fk_Chanson  
        FOREIGN KEY (numChanson) REFERENCES Chanson  
        ON DELETE CASCADE  
);
```