

Bases de Données (BD3)

Examen de Seconde Session (durée : 2 heures)

Documents autorisés : trois feuilles A4 recto-verso et personnelles. Les ordinateurs et les téléphones mobiles sont interdits.

Le barème est donné à titre indicatif.

Exercice 1 [Requêtes : 12 points]

Discogs est une base de données discographique alimentée par ses membres. On en considère ici une version ultra simplifiée **DISCOGS**, contenant les tables suivantes :

- **ALBUM** (*Num_Album, Nom_Artiste, Nom_Album, Genre, Année)
- **COPIE** (*Num_Copie, Num_Album#, Num_Vendeur#, Prix)
- **VENDEUR** (*Num_Vendeur, Pays, Frais_Port)

Les clefs primaires sont précédées d'une étoile (*) et les clefs étrangères sont suivies d'un astérisque (#). Chaque vendeur peut proposer plusieurs copies différentes (neuf ou d'occasion, en vynile, CD, cassette...) d'un même album, à des prix différents.

1. Ecrivez les requêtes SQL correspondant aux questions suivantes :

- (a) Les albums dont le genre est "Hip Hop" et dont une copie est vendue pour plus de 80 euros. (Tableau résultat : **Num_Album**).

```
SELECT A.Num_Album
FROM ALBUM A, COPIE C
WHERE A.Num_Album=C.Num_Album
AND A.Genre='Hip Hop'
AND C.Prix > 80 ;
```

- (b) Les paires d'artistes différents ayant sorti un album du même genre la même année. (Tableau résultat : **A1.Nom_Artiste, A2.Nom_Artiste**).

```
SELECT A1.Nom_Artiste, A2.Nom_Artiste
FROM Artiste A1, Artiste A2
WHERE A1.Nom_Artiste < A2.Nom_Artiste
AND A1.Genre=A2.Genre
AND A1.Année=A2.Année ;
```

- (c) Les vendeurs ne vendant que des copies d'albums de "Hip Hop". (Tableau résultat : **Num_Vendeur**).

```
SELECT Num_Vendeur
FROM VENDEUR
WHERE Num_Vendeur NOT IN
(SELECT C.Num_Vendeur
FROM ALBUM A, COPIE C
WHERE A.Num_Album=C.Num_Album
AND A.Genre<>'Hip Hop') ;
```

ou

```
SELECT V.Num_Vendeur
FROM VENDEUR V
WHERE NOT EXISTS
(SELECT *
FROM ALBUM A, COPIE C
WHERE A.Num_Album=C.Num_Album
AND C.Num_Vendeur=V.Num_Vendeur
AND A.Genre<>'Hip Hop') ;
```

- (d) Le prix moyen des copies de l'album "Runnin-N-Gunnin" (Nom_Album) de "Tommy Wright III" (Nom_Artiste), pour chaque pays dans lequel des vendeurs en proposent. (Tableau résultat : **Pays, Prix_Moyen**).

```
SELECT V.Pays, AVG(C.Prix) as Prix_Moyen
FROM ALBUM A, COPIE C, VENDEUR V
WHERE A.Num_Album=C.Num_Album
AND C.Num_Vendeur=V.Num_Vendeur
AND A.Nom_Album='Runnin-N-Gunnin'
AND A.Nom_Artiste='Tommy Wright III'
GROUP BY V.pays ;
```

- (e) Le prix le plus bas proposé pour une copie d'un album de 1984 (frais de port inclus), pour chaque vendeur dont le prix moyen des albums est inférieur à 20 euros. (Tableau résultat : **Num_Vendeur, Prix_Minimum**).

```
SELECT Num_Vendeur, MIN(C.Prix+V.Frais_Port) as Prix_Minimum
FROM ALBUM A, COPIE C, VENDEUR V
WHERE A.Num_Album=C.Num_Album
AND C.Num_Vendeur=V.Num_Vendeur
AND A.Année=1984
GROUP BY V.Num_Vendeur
HAVING AVG(C.Prix) > 20 ;
```

- (f) Les vendeurs proposant une copie d'album dont le prix est supérieur à la somme totale du prix de toutes les copies d'album qu'ils proposent pour moins de 1 euros. (Tableau résultat : **Num_Vendeur**).

```
SELECT Num_Vendeur
FROM COPIE C1
WHERE C1.Prix >
(SELECT SUM(C2.Prix)
FROM COPIE C2
WHERE C1.Num_Vendeur=C2.Num_Vendeur
AND C2.Prix < 1) ;
```

- (g) Les vendeurs qui proposent une copie de chaque album de "Hip Hop". (Tableau résultat : **Num_Vendeur**).

```
SELECT V1.Num_Vendeur
FROM VENDEUR V1
```

```

WHERE NOT EXISTS
(SELECT A1.Num_Album
FROM ALBUM A1
WHERE A1.Genre='Hip hop'
AND A1.Num_Album NOT IN
(SELECT A2.Num_Album
FROM ALBUM A2, COPIE C, VENDEUR V2
WHERE A2.Num_Album=C.Num_Album
AND V1.Num_Vendeur=V2.Num_Vendeur
AND C.Num_Vendeur=V2.Num_Vendeur) );

```

ou bien

```

SELECT C.Num_Vendeur
FROM COPIE C, Album A
WHERE A.Num_Album=C.Num_Album
AND A.Genre='Hip-Hop'
GROUP BY C.Num_Vendeur
HAVING COUNT(Distinct Num_Album)
=
(SELECT COUNT(Distinct Num_Album)
FROM ALBUM
WHERE Genre='Hip hop' );

```

Num_Album étant clef primaire de Album le second DISTINCT est d'ailleurs facultatif :

```

SELECT C.Num_Vendeur
FROM COPIE C, Album A
WHERE A.Num_Album=C.Num_Album
AND A.Genre='Hip-Hop'
GROUP BY C.Num_Vendeur
HAVING COUNT(Distinct Num_Album)
=
(SELECT Num_Album
FROM ALBUM
WHERE Genre='Hip hop' );

```

2. Ecrivez en algèbre relationnelle les requêtes correspondant aux questions (a) et (b).

•

$$\pi_{Num_Album}(\sigma_{Genre='HipHop'}(ALBUM) \bowtie \sigma_{Prix>80}(COPIE))$$

•

$$\pi_{A1.Nom_Artiste,A2.Nom_Artiste}(\sigma_{A1.Nom_Artiste \neq A2.Nom_Artiste \wedge A1.Genre=A2.Genre \wedge A1.Annee=A2.Annee}(ALBUMA1 \times ALBUMA2))$$

• La requête (c) en bonus :

$$\pi_{Num_Vendeur}(VENDEUR) - \pi_{Num_Vendeur}(VENDEUR \bowtie COPIE \bowtie \sigma_{Genre \neq 'HipHop'}(ALBUM))$$

Exercice 2 [Normalisation : 6 points]

On souhaite enrichir un peu le schéma ultra simplifié de **DISCOGS** en conservant les dépendances fonctionnelles de notre schéma initial, que l'on suppose de la forme :

- Num_Album \rightarrow Nom_Artiste, Nom_Album, Genre, Année
- Num_Copie \rightarrow Num_Album, Num_Vendeur, Prix
- Num_Vendeur \rightarrow Pays, Frais_Port

1. Etant donné ces dépendances, notre schéma initial était-il normalisé? Justifiez.

Oui, il est même en forme normale de Boyce Codd, qui est la meilleure des formes normales que nous ayons étudiées (il est donc également en 3ème forme normale, car tout schéma en BCNF est également en 3FN), car chacune des dépendances constitue une clef pour la relation à laquelle elle s'applique.

2. On souhaite maintenant raffiner l'information concernant le genre des albums en rajoutant un attribut style, de façon à ce que, par exemple, Gangsta et Trap soient des styles du genre Hip Hop. L'album Runnin-N-Gunnin appartiendra ainsi au style Gangsta, appartenant lui-même au genre Hip Hop. On ajoute afin de représenter ces contraintes les dépendances Style \rightarrow Genre et Num_Album \rightarrow Style. En partant de l'ensemble d'attributs {Num_Album, Nom_Artiste, Nom_Album, Genre, Année, Num_Copie, Num_Vendeur, Prix, Pays, Frais_Port} et des cinq dépendances fonctionnelles ci-dessus, proposez un schéma normalisé. Justifiez votre réponse. Vous pouvez utiliser un algorithme de mise en forme normale (prenez alors soin d'appliquer l'algorithme de calcul de couverture minimale) ou bien utiliser la modélisation conceptuelle si vous expliquez bien pourquoi le schéma obtenu ne mène pas à des anomalies.

On applique l'algorithme de calcul de couverture minimale, ce qui nous permet d'éliminer Num_Album \rightarrow Genre (le détail est laissé au lecteur comme exercice). On regroupe ensuite à droite tous les attributs impliqués par un même attribut, puis on élimine (Num_Album, Num_Copie) qui est inclus dans l'une des relations produites, ce qui nous donne :

- ALBUM (*Num_Album, Nom_Artiste, Nom_Album, Style, Année)
- COPIE (*Num_Copie, Num_Album#, Num_Vendeur#, Prix)
- VENDEUR (*Num_Vendeur, Pays, Frais_Port)
- STYLE (*Style, Genre)

Le schéma obtenu est en 3NF, mais également en BCNF.

3. La vraie base de données Discogs contient des informations sur les acheteurs, ainsi que sur les adresses des acheteurs et vendeurs. On suppose qu'un groupe de vendeurs décide à des fins commerciales d'enrichir le schéma de Discogs en ajoutant une nouvelle table **CIBLE** (*Num_Acheteur#, *Type_Vendeur#, Num_Vendeur#). Ce groupe de vendeurs cherche ainsi à répertorier pour chaque acheteur et pour chaque type de vendeur, le vendeur de ce type se trouvant géographiquement le plus prêt de l'acheteur en question (le type étant un attribut défini par le genre dans lequel est spécialisé le vendeur : e.g., Hip Hop, Soul / Funk ou bien encore Rock). Ce groupe de vendeurs a en tête les deux dépendances fonctionnelles Num_Acheteur, Type_Vendeur \rightarrow Num_Vendeur et Num_Vendeur \rightarrow Type_Vendeur. Cette nouvelle relation est-elle en forme normale de Boyce Codd? Est-elle en troisième forme normale? Justifiez votre réponse.

Non, cette relation n'est pas en BCNF. Le problème vient de la seconde DF, Num_Vendeur \rightarrow Type_Vendeur, dont le membre gauche, Num_Vendeur, n'est pas une

clef pour la relation (alors que Num_Acheteur, Type_Vendeur celui de la première DF, Num_Acheteur, Type_Vendeur -> Num_Vendeur, en est bien une). En revanche, le membre droit de cette DF, Type_Vendeur, est primaire (rappel : un attribut est primaire s'il appartient à une clef candidate, ce qui est bien le cas ici, de par la première DF). La relation est donc en 3FN.

Exercice 3 [Information Incomplète : 2 points]

On considère une instance de la table ALBUM de la forme suivante :

ALBUM	Num_Album	Nom_Artiste	Nom_Album	Genre	Année
	42	Wu-Tang Clan	Enter The Wu-Tang (36 Chambers)	Hip Hop	1993
	69	Mykki Blanco	Cosmic Angel : The Illuminati Prince/ss		2012
	555	Five Finger Posse	Hood Goth	Hip Hop	2016
	3	Mason vs. Princess Superstar	Perfect (Exceeder)	Electro	2006

1. Donnez le résultat des requêtes suivantes :

- (a)

```
SELECT Nom_Album
FROM ALBUM
WHERE Genre <> Electro ;
```

Les seuls albums dont le genre est ici différent de l'electro sont les albums de hip hop, en effet un null n'est jamais ni égal, ni différent de quelque valeur que ce soit (comme expliqué en cours, la valeur de vérité de la comparaison est indéterminée). Le résultat est donc Enter The Wu-Tang (36 Chambers) et Hood Goth.

- (b)

```
SELECT Nom_Album
FROM ALBUM
WHERE Nom_Album NOT IN
(SELECT Nom_Album
FROM Album
WHERE Genre <> Electro) ;
```

On cherche ici les albums qui ne sont pas dans le résultat de la requête précédente, ie Cosmic Angel : The Illuminati Prince/ss et Perfect (Exceeder).

2. Auriez-vous obtenus les mêmes résultats si le genre de l'album "Cosmic Angel : The Illuminati Prince/ss" avait été "Hip Hop"? Justifiez votre réponse.

Non, dans le premier cas on aurait obtenu Enter The Wu-Tang (36 Chambers), Hood Goth et Cosmic Angel : The Illuminati Prince/ss. Dans le second cas on aurait obtenu Perfect (Exceeder). (Réponse justifiée dans la question précédente, point (a).)