

Informatique TSI2. DS01

Correction

EXERCICE 01

La bibliothèque francophone d'Arkham utilise une base de données pour gérer les emprunts. La table *inscrits* contient cinq colonnes : *Nom*, *Prenom*, *Naissance*, *DateCotisation*, *Numero*. Voici quelques lignes de la table :

<i>inscrits</i>				
Nom	Prenom	Naissance	DateCotisation	Numero
LAMBERT	Bernard	1985-03-22	2013-08-19	7
RICE	Warren	1975-04-03	2014-09-21	11
VASSEUR	Jade	1995-05-17	2014-02-15	12
⋮	⋮	⋮	⋮	⋮

La table des *livres* indique *Titre*, *auteur*, *CoteClassement* de chaque livre. Voici quelques lignes de la table (JD sont les initiales de John Doe ou Jane Doe) :

<i>livres</i>		
Titre	Auteur	CoteClassement
Unaussprechlichen Kulten	Friedrich Wilhelm von Junzt	FWJ01
Le Roi en Jaune	Robert William Chambers	RWC03
De Vermis Mysteriis	Inconnu	JD01
Le culte des goules	Comte Francoise Honoré Balfour d'Erlette	FHBE01
⋮	⋮	⋮

Nous avons enfin une table des prêts, nommée *prets*. Ses colonnes sont *NumeroInscription*, *DatePret*, *RENDU*. Un prêt est une relation entre un inscrit et un livre. Voici quelques lignes de la table :

<i>prets</i>			
NumeroInscription	Cote	DatePret	RENDU
2	FHBE01	2008-10-01	TRUE
2	FHBE01	2014-01-09	TRUE
11	JD01	2011-11-05	FALSE
11	RWC03	2013-11-05	TRUE
⋮	⋮	⋮	⋮

Q1. Une clé primaire de *inscrits* est *inscrits.Numero*, une clé primaire de *livres* est *livres.CoteClassement*, deux clés étrangères de *prets* sont *Prets.NumeroInscription* et *Prets.Cote*.

Q2. En français ce que signifie les quatre lignes visibles de la table *prets* : L'inscrit numéro 2 a pris le livre Le Culte des Goules deux fois, une fois le 01/10/2008 et la seconde fois le 09/01/2014 et la rendu les deux fois, l'inscrit numéro 11 qui est Warren Rice a pris le livre De Vermis Mysteriis le 05/11/2011 et ne l'a jamais rendu car il était trop bien et il a emprunté le 05/11/2013 Le Roi en Jaune et l'a rendu par contre.

Q3. Requête SQL qui renvoie la liste des cotes des livres prêtés non rendus :
 SELECT Cote FROM prets WHERE Rendu=False;

Q4. Requête SQL qui renvoie le nombre d'inscrits dans la bibliothèque :
 SELECT COUNT(*) FROM inscrits;

Q5. Requête SQL qui renvoie la liste des noms des livres de la bibliothèque classés par ordre décroissant de leur CoteClassement. (On suppose que SQL gère l'ordre lexicographique avec un mélange de lettres et de chiffres.)

SELECT livres.Titre FROM livres ORDER BY livres.CoteClassement DESC;

Q6. Requête SQL qui renvoie le nombre d'inscrits dans la bibliothèque ayant déjà emprunté au moins un livre. Il faut compter combien d'inscrits apparaissent dans la table *prets* en évitant les doublons. On sent DISTINCT approcher.

```
SELECT COUNT(DISTINCT NumeroInscription) FROM prets ;
```

Q7. Requête SQL qui renvoie la liste des livres déjà empruntés par Roland FRANKLIN (Roland est le prénom et on suppose que Roland FRANKLIN n'a pas d'homonyme).

Indication : Il est temps de sortir les JOIN

```
SELECT livres.CoteClassement FROM livres JOIN prets ON livres.CoteClassement = prets.Cote
JOIN inscrits ON prets.NumeroInscription = inscrits.Numero WHERE inscrits.Nom = " Franklin"
AND Prenom = "Roland" ;
```

EXERCICE 02

La société Golden Web Pizza permet de se faire livrer chez soi une pizza parmi un catalogue immense. La société gère ses commandes via une base de données. Il y a quatre tables.

La table *pizza* contient 2 champs : *nom-pizza*, *prix-euro*

Attention, le prix de chaque pizza est celui de 2014. Voici quelques lignes de la table :

<i>pizza</i>	
nom-pizza	prix-euro
Margherita	8
Caviar-totale	70
⋮	⋮

La table *clients* contient 4 champs : *Nom*, *Prenom*, *Adresse*, *numero-client*,
Voici quelques lignes de la table :

<i>clients</i>			
Nom	Prenom	Adresse	numero-client
BLANCHARD	Nathan	7 rue du mail, 75002 Paris	42
FABRE	Lucas	18 rue de l'électricité, 21000 Dijon	47
⋮	⋮	⋮	⋮

La table *commande* contient cinq colonnes : *num-client*, *nom-pizza*, *prix-euro*, *num-order*, *date-order* Voici quelques lignes de la table :

<i>commande</i>				
num-client	nom-pizza	prix-euro	num-order	date-order
42	Caviar-totale	68	1024	1999-08-19
47	Margherita	8	16384	2014-12-12
47	Super-champignons	13	16384	2014-12-12
⋮	⋮	⋮	⋮	⋮

La table *ingredients* contient 2 colonnes : *nom-pizza*, *ingredient*
Voici quelques lignes de la table :

<i>ingredients</i>	
nom-pizza	ingredient
Margherita	purée de tomate
Margherita	preparation fromagère
Caviar-totale	caviar
Caviar-totale	foie gras
Caviar-totale	crème fraîche
⋮	⋮

Q1. Une clé primaire est *numero-client* de la table *clients* et une clé étrangère est *num-client* de la table *commande*

Q2. Comment interpréter en français la première ligne de données de la table *commande* ? Nathan Blanchard le client numéro 42 commande une pizza Caviar-totale le 19/08/1999 au prix de 68 euros et c'est la commande numéro 1024.

Q3. Comment interpréter en français les deuxième et troisième ligne de données de la table *commande* ? Le client Lucas Fabre a commandé le même jour le 12/12/2014 (car il a une grosse faim) deux pizzas différentes (Margherita et Super-champignons) au prix total de $8 + 13 = 21$ euros.

Q4. On peut déduire de ces tables que le nombre d'ingrédients de la pizza Margherita est au moins de 2 et de la pizza Caviar-totale au moins de 3.

Q5. Cette base de données permet-elle pour un client d'avoir deux fois la même pizza (par exemple deux Margherita) dans la même commande ? La réponse est non car cela crée deux lignes identiques dans la table *commande*.

Q6 Il y a un intérêt de mettre le prix dans la table des commandes vu qu'il est déjà dans la table des pizzas. En effet le prix évolue selon les années et les prix de la table *pizzas* sont ceux de 2014 seulement.

Q7. Requête SQL qui donne le total des ventes en euros de pizza le jour 2014-10-18 :

SELECT SUM(commande.prix-euro) FROM commande WHERE commande.date-order="2014-10-18";

Q8. Requête SQL qui donne la liste des noms de pizzas contenant les deux ingrédients : « purée de tomate » et « tomate »

```
SELECT DISTINCT(nom-pizza) FROM pizza JOIN ingredients
ON pizza.nom-pizza=ingredients.nom-pizza
WHERE ingredient="tomate" OR ingredient="puree de tomates";
```

Q9. Un client est allergique à la tomate. Les deux seuls ingrédients contenant de la tomate sont « purée de tomate » et « tomate ». Écrivons une requête SQL qui renvoie la table des pizzas (avec leur nom et leur prix) ne contenant pas ces ingrédients ? On pourra utiliser EXCEPT et le résultat de la question précédente.

```
(SELECT * FROM pizza)
EXCEPT
(SELECT DISTINCT(nom-pizza) FROM pizza JOIN ingredients ON pizza.nom-pizza=ingredients.nom-pizza
WHERE ingredient="tomate" OR ingredient="puree de tomates");
```

Q10. Requête SQL qui renvoie la liste des pizzas commandées entre le 2014-01-01 et le 2014-12-31 (les répétitions sont autorisées) :

```
SELECT pizza.* FROM pizza JOIN commande ON pizza.nom-pizza=commande.nom-pizza WHERE
commande.date-order >= 2014-01-01 AND commande.date-order <= 2014-12-31;
```

Q11. Requête SQL qui renvoie pour chaque client la somme en euro qu'il a rapporté à la société sur toutes les années.

Indication : chaque client est défini par son numéro client. On renommera depense la somme en euro dépensée par chaque client et on pourra utiliser GROUP BY.

```
SELECT commande.num-client,SUM(commande.prix-euro) AS depense FROM commande GROUP BY
commande.num-client;
```