Université de Caen Normandie
Département de Mathématiques et Informatique
Licence informatique L2 - 2022-2023

Bases de données
TD 5 et 6 (SQL et algèbre relationnelle) – TP 3 (SQL)

Indication :
— TD 1 : exercices 1 à 3
— TD 2 : exercices 3 à 6

Exercice 1

On s'intéresse à la base de données formée des relations ACTEUR(Nom), REALISATEUR(Nom) et PRODUCTEUR(Nom). On souhaite écrire une requête qui donne les noms des acteurs qui sont aussi réalisateurs ou producteurs ou sont les trois. Il est possible que l'une (ou plusieurs) des tables ACTEUR, REALISATEUR ou PRODUCTEUR soit vide.

1. Expliquez pourquoi la requête suivante ne donne pas le résultat attendu dans certains cas. Quels sont ces cas ?

```
SELECT A.Nom
FROM ACTEUR A, REALISATEUR R, PRODUCTEUR P
WHERE A.Nom = R.Nom
OR A.Nom = P.Nom;
```

2. Écrivez deux requêtes SQL qui fournissent le résultat attendu dans *tous* les cas.

Exercice 2 - Est-il possible d'utiliser `NATURAL JOIN` ?

On introduit la nouvelle table :
REDUCTION(Code, Quantite, Reduction)
qui indique, pour le code d'un produit, la quantité minimale de ce produit à acheter pour avoir une réduction et le pourcentage de réduction accordé sur ce produit si la réduction s'applique.
Par exemple, si on a Reduction = 9.5 et Quantite=50 pour le produit CL45, cela signifie que l'on bénéficie d'une réduction du prix de 9.5% si on commande CL45 avec une quantité supérieure ou égale à 50. Pour savoir si une réduction s'applique, on considère toujours l'achat d'un produit pour une seule commande.
La clé primaire de REDUCTION est Code.

Exemple d'extension de la table REDUCTION :

Code	Quantite	Reduction
CH262	0	0
CH264	200	7
CH464	100	12.2
CL45	50	9.5
CL60	10	6.5
PL222	120	7.1
PL224	100	7.12

Partie 1 : Écrivez l'instruction qui permet de créer la table REDUCTION.

Partie 2 : Exprimez chacune des requêtes suivantes par une instruction MySQL.

1. Quantité totale de chevilles commandées par des clients de Toulouse (c'est-à-dire, la somme des quantités de chevilles commandées par l'ensemble des clients de Toulouse).
2. Pour chaque produit commandé, nombre de commandes où ce produit a été commandé avec une réduction non nulle.
   Rappel : il faut que la commande porte sur une quantité supérieure ou égale à la quantité minimale pour avoir la réduction.
3. Que retourne la requête :

   ```
   SELECT *
   FROM REDUCTION NATURAL JOIN DETAIL ;
   ```

   Pourquoi ?

Exercice 3

Dans la suite, on travaille avec la base de données `Commandes` (vue en cours) et constituée des relations :
CLIENT(RefC, NomC, Ville)
PRODUIT(RefP, TypeP, Prix)
COMMANDE(RefCom, RefC, DateCom)
DETAIL(RefCom, RefP, Quantité)

Exprimez chacune des requêtes suivantes par une instruction SQL.

1. Pour chaque type de produit, les noms des clients de Toulouse qui ont commandé un produit de ce type. Le résultat doit être présenté suivant l'ordre alphabétique des types des produits, et pour chaque type de produit, suivant l'ordre alphabétique des noms des clients. La table résultat ne doit pas contenir deux fois la même ligne.
2. Produits de type Clou qui n'ont pas été commandés par certains clients (autrement dit, produits de type Clou pour chacun desquels il existe un client qui ne l'a pas commandé).
3. Noms des clients qui n'ont commandé aucun des produits commandés par Vanderka.
   Indication : Reformulez la requête : il s'agit de tous les clients sauf ceux qui ont commandé au moins un des produits commandés par Vanderka.
4. Moyenne du montant total d'une commande, en considérant toutes les commandes.
   Indication : Utilisez une requête imbriquée dans le FROM. Que représente cette sous-requête ?
5. Références de la (ou des) commande(s) dont le montant total est maximal parmi toutes les commandes. On donnera aussi ce montant.
   Indication : Utilisez le ALL.

Exercice 4

Calculez la table R/S, où R et S sont les tables définies par :

R :

A	B	C	D
a	b	c	d
e	d	e	f
a	b	d	e
e	d	c	d
b	c	e	f
a	b	e	f

S :

C	D
c	d
e	f

Exercice 5

Exprimez chacune des requêtes suivantes par une expression de l'algèbre relationnelle.

1. Donner pour chaque référence de commande les références de produits qu'elle ne contient pas.
2. Donner les références de commandes qui ne contiennent pas certains produits.
3. Donner les références de commandes qui contiennent tous les produits.

Exercice 6

1. Quelle requête exprime l'expression suivante ? (où R/S désigne la division de la relation R par la relation S)

   ((CLIENT ⋈ COMMANDE ⋈ DETAIL ⋈ PRODUIT)[Ville,TypeP])
   / (PRODUIT[TypeP])

2. Traduisez cette requête en SQL.

TP 3

La base de données sur laquelle on travaille est constituée des relations :

CLIENT(RefC, NomC, Ville, CAT),
L'attribut CAT qui indique la catégorie du client, sa valeur peut être NULL (ce qui signifie qu'elle n'est pas renseignée).

PRODUIT(RefP, TypeP, Prix, QStock),
QStockl donne, pour chaque produit, la quantité disponible en stock.

COMMANDE(RefCom, RefC, DateCom)

DETAIL(RefCom, RefP, Quantité).
La clé (PRIMARY KEY) de DETAIL est composée des deux attributs RefCom et RefP.

Exercice 1 - Est-il possible d'utiliser `NATURAL JOIN` ?

Cet exercice a été préparé lors du TD 5.

On introduit la nouvelle table :
`REDUCTION (Code, Quantite, Reduction)`
qui indique, pour le code d'un produit, la quantité minimale de ce produit à acheter pour avoir une réduction et le pourcentage de réduction accordé sur ce produit si la réduction s'applique.
Par exemple, Reduction = 9.5 et Quantite=50 pour le produit CL45 signifie que l'on bénéficie d'une réduction du prix de 9.5% si on commande CL45 avec une quantité supérieure ou égale à 50. Pour savoir si une réduction s'applique, on considère toujours l'achat d'un produit pour une seule commande.
La clé primaire de `REDUCTION` est `Code`.

Exemple d'extension de la table `REDUCTION` :

Code	Quantite	Reduction
CH262	0	0
CH264	200	7
CH464	100	12.2
CL45	50	9.5
CL60	10	6.5
PL222	120	7.1
PL224	100	7.12

Partie 1 :
— Écrivez l'instruction qui permet de créer la table `REDUCTION`.
— Insérez dans la table `REDUCTION` les lignes de l'extension donnée en exemple en utilisant le fichier de données `reduction.dat` que vous téléchargerez sur ecampus.

Partie 2 : Exprimez chacune des requêtes suivantes par une instruction MySQL.

1. Quantité totale de chevilles commandées par des clients de Toulouse (c'est-à-dire, la somme des quantités de chevilles commandées par l'ensemble des clients de Toulouse).
   Peut-on utiliser `NATURAL JOIN` ? Pourquoi ?

2. Code des produits ayant une réduction non nulle avec leur pourcentage de réduction classés par ordre croissant de réduction.
3. Pour chaque produit commandé, nombre de commandes où ce produit a été commandé avec une réduction non nulle.
   Rappel : il faut que la commande porte sur une quantité supérieure ou égale à la quantité minimale pour avoir la réduction.
   Que se passe-t-il si on utilise `NATURAL JOIN` ? Pourquoi ?
4. Que retourne la requête :

   ```
   SELECT *
   FROM REDUCTION NATURAL JOIN DETAIL ;
   ```

   Pourquoi ?

Exercice 2 - Mise à jour de n-uplets

Dans l'extension de la table CLIENT du TP1, la valeur de l'attribut CAT n'est pas définie pour les clients Mercier et Neuman. On suppose maintenant que Mercier est dans la catégorie C1 et Neuman dans la catégorie C2. Mettez à jour la table CLIENT.

Exercice 3 - Requêtes

Exprimez chacune des requêtes suivantes par une instruction SQL.

1. Références et noms des clients qui ont chacun commandé plus de 50 clous, en considérant toutes les commandes de chaque client.
2. Noms des clients qui n'ont commandé aucun des produits commandés par Vanderka.
   Indication : Reformulez la requête : il s'agit de tous les clients sauf ceux qui ont commandé au moins un des produits commandés par Vanderka.
3. Moyenne du montant total d'une commande, en considérant toutes les commandes.
   Indication : Utilisez une requête imbriquée dans le `FROM`. Que représente cette sous-requête ?
   La sous-requête dans le `FROM` calcule la somme de chaque commande.
4. Références de la (ou des) commande(s) dont le montant total est maximal parmi toutes les commandes. On donnera aussi ce montant.
   Indication : Utilisez le `ALL`.
5. Noms des clients qui ont commandé le produit CL45 ou le produit CL60, mais pas les deux. *Indication :* Ces clients n'ont commandé qu'un seul produit.
6. Noms des clients qui ont commandé exactement deux des trois produits CL45, CL60 et CH464.
7. Pour chaque client (représenté par sa référence), le nombre total de clous qu'il a commandés (en considérant toutes les commandes). Ne pas oublier les clients qui n'ont commandé aucun clou.
8. Pour chaque ville et chaque catégorie de clients de cette ville, la somme, en euros, des montants de toutes les commandes des clients habitant cette ville et appartenant à cette catégorie. Pour une ville donnée, attention à ne pas oublier les catégories où personne n'a fait de commande !
9. Villes où on trouve toutes les catégories de clients.
   On donnera deux solutions (l'une avec `COUNT`, l'autre avec un double `NOT IN`).