

Chapitre 8

QCM

1. B.

2. C.

3. A.

4. A.

5. B.

6. A. ET C.

7. A. ET B.

8. B. ET C.

9. A. ET B.

10. A. ET B.

11. B. Il n'existe qu'une seule clause WHERE, les autres conditions étant reliées par des clauses AND.

12. B. C'est HAVING, la clause WHERE n'appliquant une restriction que sur des enregistrements et non sur des agrégats.

13. B. On peut effectuer une requête d'insertion partielle.

14. B. Elle supprime l'ensemble des enregistrements de la table, aucune clause WHERE de restriction n'étant spécifiée.

15. B. L'option Cascade, qui permet de supprimer des enregistrements en cascade, ne fonctionne que parce qu'il y a des contraintes d'intégrité référentielle entre attributs, et non du fait que les attributs soient nommés de manière identique ou non.

Exercices

1. Interrogation de la base de données *Les comptoirs*

Écrire les requêtes SQL suivantes :

- a. Liste des produits dont le stock est inférieur au niveau de réapprovisionnement :

```
SELECT [Nom du produit], [Unités en stock],  
[Niveau de réapprovisionnement]  
FROM Produits  
WHERE [Unités en stock] < [Niveau de  
réapprovisionnement];
```

- b. Noms des sociétés et numéros de téléphone des fournisseurs non français triés par société :

```
SELECT Société, Téléphone  
FROM Fournisseurs  
WHERE Pays <> "France"  
ORDER BY Société;
```

- c. Noms des sociétés des clients dont le nom commence par la lettre M :

```
SELECT Société  
FROM Clients  
WHERE Société LIKE "m*";
```

- d. Liste des fournisseurs (Société, Ville, Fax) qui ont un fax :

```
SELECT Société, Ville, Fax  
FROM Fournisseurs  
WHERE Fax IS NOT NULL
```

- e. Liste des produits dont le stock est compris entre 10 et 30, trié par unités en stock :

```
SELECT [Nom du produit], [Unités en stock]  
FROM Produits  
WHERE [Unités en stock] BETWEEN 10 AND 30  
ORDER BY [Unités en stock] DESC;
```

2. Interrogation avancée de la base de données *Les comptoirs*

Écrire les requêtes SQL suivantes :

- a. Liste des produits (nom, société de fournisseur) classée par société du fournisseur et par produit :

```
SELECT [Nom du produit], Société  
FROM Fournisseurs, Produits  
WHERE Produits.[N° fournisseur] =  
Fournisseurs.[N° fournisseur]  
ORDER BY Société, [Nom du produit];
```

CORRIGÉ

b. Nom des clients (société, pays) n'ayant jamais passé commande :

```
SELECT Société, Pays
FROM Clients
WHERE [Code client] NOT IN (select [Code
client] from commandes);
```

c. Liste des produits (nom du produit, société du fournisseur) achetés au fournisseur Tokyo Traders :

```
SELECT [Nom du produit], Société
FROM Fournisseurs, Produits,
WHERE Produits. [N° fournisseur] = Fournisseurs.[N°
fournisseur]
AND Société="Tokyo Traders ";
```

d. Produits, société et pays des fournisseurs de catégorie Condiments :

```
SELECT [Nom de produit], Société, Pays, [Nom de
catégorie]
FROM Fournisseurs, Produits, Catégories,
WHERE Produits. [N° fournisseur] =
Fournisseurs.[N° fournisseur]
AND Produits.[Code catégorie]= Catégorie.[Code
catégorie]
AND [Nom de catégorie]=" Condiments ";
```

e. Produits, société et pays des fournisseurs de catégorie Condiments issus du Japon ou de Singapour :

```
SELECT [Nom de produit], Société, Pays, [Nom de
catégorie]
FROM Fournisseurs, Produits, Catégories,
WHERE Produits. [N° fournisseur] = Fournisseurs.[N°
fournisseur]
AND Produits.[Code catégorie]= Catégorie.[Code
catégorie]
AND [Nom de catégorie]=" Condiments ";
AND (Pays="Japon" OR Pays="Singapour")
```

3. Interrogation approfondie de la base de données *Les comptoirs*

Écrire les requêtes SQL suivantes :

a. Montant du salaire moyen en vigueur dans l'entreprise :

```
SELECT Avg([salaire annuel actuel])
AS [Salaire moyen général]
FROM Employés;
```

b. Nombre de clients par pays :

```
SELECT Count(*) AS [Nb Clients], pays
FROM clients
GROUP BY Pays
```

c. Nombre de clients par pays pour les pays qui comptent au moins de dix clients :

```
SELECT Count(*) AS [Nb Clients], pays
```

CORRIGÉ

```
FROM clients
GROUP BY Pays
HAVING Count(*) >= 10
```

d. Nombre de commandes, chiffre d'affaires total et montant moyen d'une commande :

```
SELECT Count(*) AS [Nb de Commandes],
SUM(Montant) AS [Chiffres d'affaires],
AVG(Montant) AS [Moyant moyen commande]
FROM Commandes
```

**e. Nombre de commandes, chiffre d'affaires total et montant moyen d'une commande pour l'année 1990.
Deux versions sont attendues : une avec l'opérateur LIKE et une avec une fonction de DATE.**

Version avec LIKE :

```
SELECT Count(*) AS [Nb de Commandes], SUM(Montant) AS [Chiffres d'affaires], AVG(Montant)
AS [Moyant moyen commande]
FROM Commandes
WHERE [Date Commande] LIKE "**/**/1990"
```

Version avec Fonction de date :

```
SELECT Count(*) AS [Nb de Commandes], SUM(Montant) AS [Chiffres d'affaires], AVG(Montant)
AS [Moyant moyen commande]
FROM Commandes
WHERE YEAR ([Date Commande]) = 1990
```

Cas de synthèse

ADAPTATION DE LA BASE DE DONNÉES DE LA PIZZÉRIA BONAPIZZA

Mission n° 1 : Interrogation de la base de données

1.1 Expliquer pourquoi ce résultat n'est pas optimisé.

Il y a des doublons.

1.2 Modifier la requête écrite par le gérant, de façon à améliorer son résultat.

```
SELECT DISTINCT Ville_Cli FROM Clients;
```

1.3 Écrire la requête d'extraction de données permettant d'obtenir ce résultat.

```
Select ville_Cli, count(*) as NbCli  
From CLIENTS  
Group by ville_Cli
```

1.4 Écrire la requête d'extraction de données permettant de répondre à cette demande.

```
Select ville_Cli, count(*) as NbCli  
From CLIENTS  
Group by ville_Cli  
Having cout(*) > 2
```

1.5 Écrire la requête d'extraction de données permettant d'obtenir le libellé de la pizza la moins chère.

```
Select Design_Pizza  
From PIZZAS  
WHERE Prix = (Select MIN(Prix) From PIZZAS)
```

1.6 Le gérant aurait besoin d'informations concernant les commandes. Il aimerait connaître pour chaque commande :

a. la référence, la désignation, la quantité commandée et le prix de chaque pizza, ainsi que le montant total de la commande.

```
Select No_Cde, Ref_Pizza, Design_Pizza, Prix, Qte_Cde, SUM (Prix * Qte_Cde) AS [Montant  
total]  
From PIZZAS, LIGNES_COMMANDES, COMMANDE  
WHERE PIZZAS.Ref_Pizza = LIGNES_COMMANDES.Ref_Pizza  
AND LIGNES_COMMANDES.No_Cde = COMMANDES.No_Cde
```

b. le nom et le prénom du client, ainsi que le total à payer.

```
Select No_Cde, Nom_Cli, Pren_Cli, SUM (Prix * Qte_Cde) AS [Total à payer]  
From PIZZAS, LIGNES_COMMANDES, COMMANDE, CLIENTS  
WHERE PIZZAS.Ref_Pizza = LIGNES_COMMANDES.Ref_Pizza  
AND LIGNES_COMMANDES.No_Cde = COMMANDES.No_Cde  
AND CLIENTS.No_Cli = COMMANDES.No_Cli
```

CORRIGÉ

Mission n° 2 : Mise à jour de la base de données

2.1 Fournir la requête permettant de corriger cette erreur dans la base de données.

```
Update LIGNES_COMMANDES set QteCde = 4 where No_Cde = 3 and RefPizza = "NAP1"
```

2.2 Écrire, dans l'ordre d'exécution, les requêtes de mise à jour des données à mener pour prendre en compte ces événements.

```
Insert into CLIENTS Values (6, "Jaoui", "Irène", "10 rue de l'Ortolan", "06160", "Antibes", NULL)
```

```
Insert into COMMANDES Values (9, #29/10/2020#, 6)
```

```
Insert into LIGNES_COMMANDES Values (9, FRU2, 5)
```

2.3 Justifier pourquoi une seule requête est suffisante pour supprimer la commande dans sa totalité.

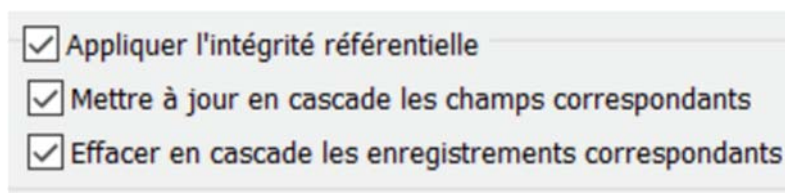
Seule une requête de suppression sur la table *Commandes* est nécessaire.

En effet, l'intégrité référentielle entre *Lignes_Commandes* et *Commandes* garantit la cohérence des données grâce à la propagation en *Cascade* dans la table *Lignes_Commandes* de toute mise à jour effectuée sur la table *Commandes*.

2.4 Décrire la manipulation à effectuer sur la base de données pour permettre la suppression en une requête.

Dans le paramétrage des liens entre tables, sur le lien entre les tables *Lignes_Commandes* et *Commandes*, il faut cocher 'option « Effacer en cascade ».

Figure 1



2.5 Fournir la requête permettant de prendre en compte cet événement dans la base de données.

```
DELETE from COMMANDES WHERE No_Cde = 3
```

Mission n° 3 : Export de données

3.1 Proposer une solution pour répondre à sa demande.

Il faut exporter les données depuis la base de données dans un fichier « plat » (au format texte) de type CSV ou XML, puis importer le fichier généré dans le tableur du gérant.