



Conception de Base de Données:
TP1 Partie 2 - Exercices

SANNA Thomas

November 26, 2024

Contents

1 Question 1	3
1.1 Modèle Conceptuel de Données (MCD)	3
1.2 Couverture Minimale	4
2 Question 2	5
2.1 Afficher tous les clients qui habitent Corte ou Omessa	5
2.1.1 Algèbre Relationnelle	5
2.1.2 SQL	5
2.2 Afficher les noms et villes des clients	5
2.2.1 Algèbre Relationnelle	5
2.2.2 SQL	6
2.3 Afficher les noms et téléphones des Clients habitant Porto-Vecchio	6
2.3.1 Algèbre Relationnelle	6
2.3.2 SQL	7
2.4 Afficher les noms de clients ayant passé une commande le 10/11/2020	7
2.4.1 Algèbre Relationnelle	7
2.4.2 SQL	8
2.5 Afficher les produits achetés par le client 102	9
2.5.1 Algèbre Relationnelle	9
2.5.2 SQL	9
2.6 Afficher les produits achetés par le client "Delhom"	10
2.6.1 Algèbre Relationnelle	10
2.6.2 SQL	10
2.7 Noms des clients ayant acheté du Chocolat	11
2.7.1 Algèbre Relationnelle	11
2.7.2 SQL	11
2.8 Noms des clients n'ayant jamais acheté de lait	12
2.8.1 Algèbre Relationnelle	12
2.8.2 SQL	13
2.9 Noms des clients ayant acheté des produits dont le prix est supérieur à 4 Euros	13
2.9.1 Algèbre Relationnelle	13
2.9.2 SQL	14
2.10 Liste des produits achetés par des porto-vecchiais	15
2.10.1 Algèbre Relationnelle	15
2.10.2 SQL	15
3 Question 3	17
3.1 Création de la base de données	17
3.1.1 Montant total des commandes du client 102	17
3.1.2 Nombre total de commandes du client 102	18

1 Question 1

1.1 Modèle Conceptuel de Données (MCD)

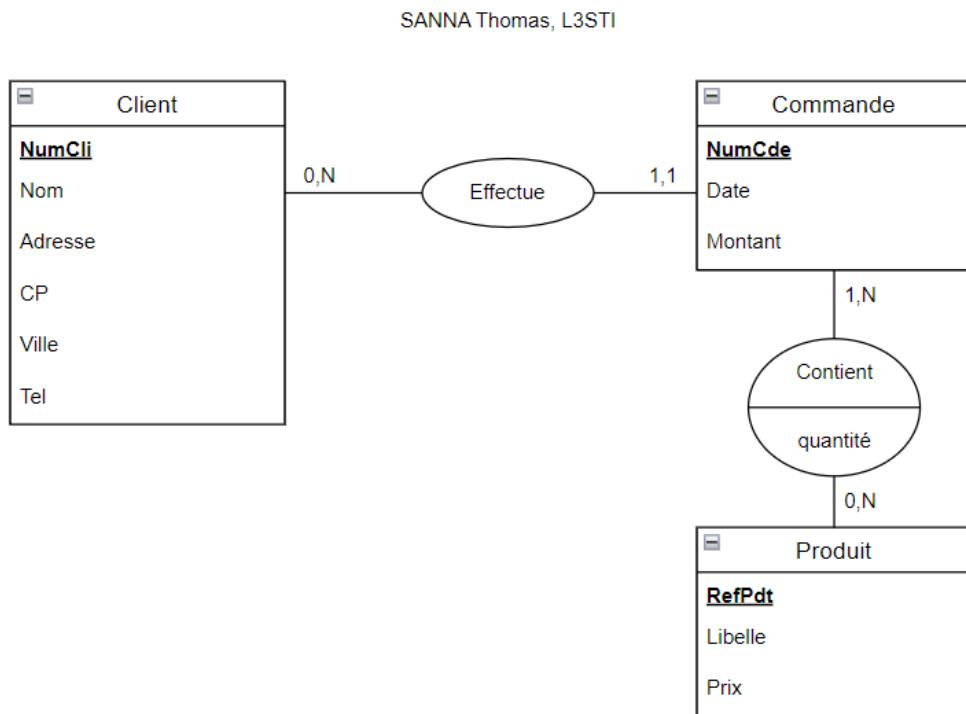


Figure 1: Modèle Conceptuel de Données, en utilisant Draw.io.

1.2 Couverture Minimale

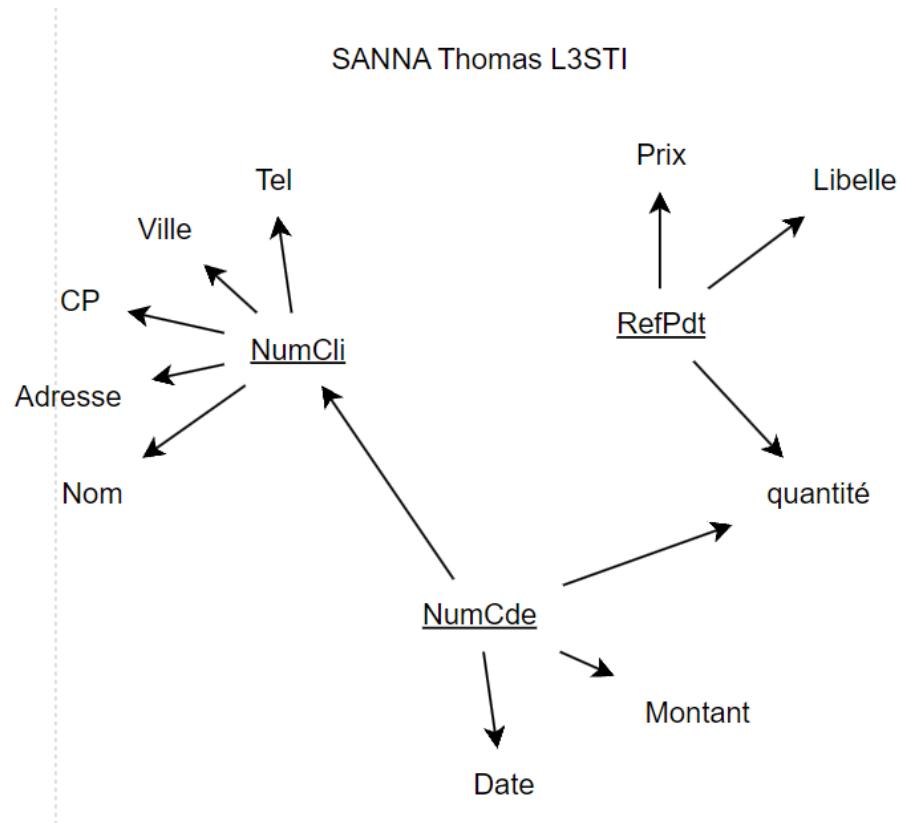
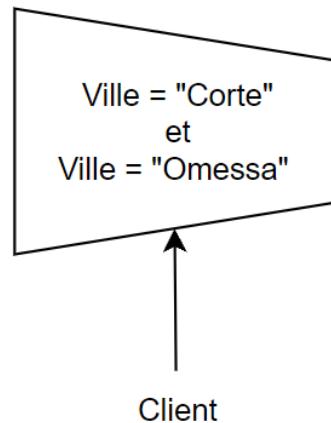


Figure 2: Couverture Minimale, en utilisant Draw.io.

2 Question 2

2.1 Afficher tous les clients qui habitent Corte ou Omessa

2.1.1 Algèbre Relationnelle

$$\sigma_{Ville = "Corte"} \vee Ville = "Omessa" (Client)$$


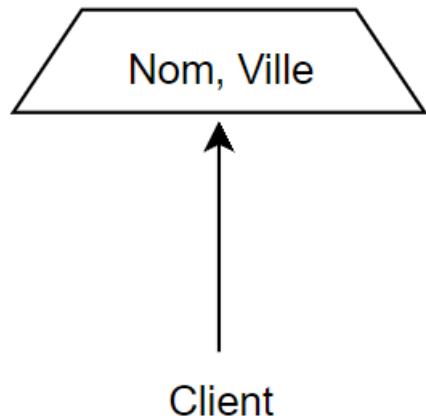
2.1.2 SQL

```
SELECT *
FROM Client
WHERE Ville = 'Corte' OR Ville = 'Omessa';
```

2.2 Afficher les noms et villes des clients

2.2.1 Algèbre Relationnelle

$$\pi_{Nom, Ville} (Client)$$



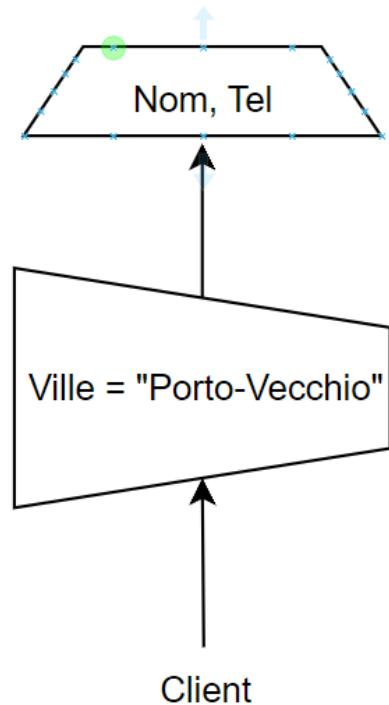
2.2.2 SQL

```
SELECT Nom, Ville  
FROM Client;
```

2.3 Afficher les noms et téléphones des Clients habitant Porto-Vecchio

2.3.1 Algèbre Relationnelle

$$\begin{aligned} R_1 &= \sigma_{Ville = "Porto-Vecchio"}(Client) \\ R_2 &= \pi_{Nom, Tel}(R_1) \end{aligned}$$



2.3.2 SQL

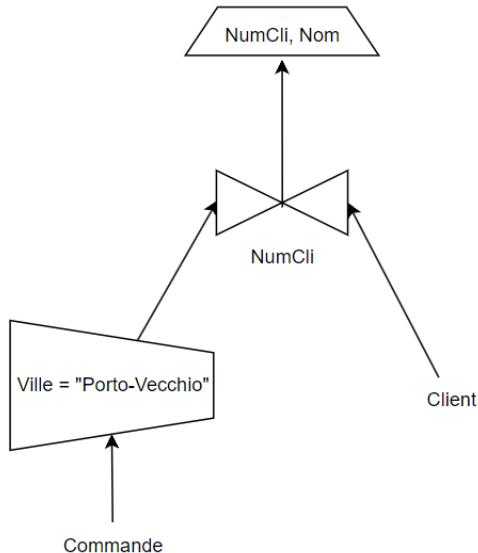
```

SELECT Nom, Tel
FROM Client
WHERE Ville = 'Porto-Vecchio';
  
```

2.4 Afficher les noms de clients ayant passé une commande le 10/11/2020

2.4.1 Algèbre Relationnelle

$R_1 = \sigma_{Date = "10/11/2020"}(Commande)$
 $R_2 = (R_1) \Delta_{\begin{matrix} NumCl \\ NomCl \end{matrix}} (Client)$
 $R_3 = \pi_{NumCl, Nom}(R_2)$
 (NB : J'ai préféré mettre ~~NumCl~~ avec les noms pour plus de cohérences. Sinon, on aurait des doublons.)



NB : Pour les prochaines questions qui le demandent, j'ai aussi décidé de mettre NumCli en plus de Nom, pour plus de clarté et de précision.

2.4.2 SQL

```
SELECT c.NumCli, c.Nom
FROM Client c
JOIN Commande co ON c.NumCli = co.NumCli
WHERE co.Date = '2020-11-10';
```

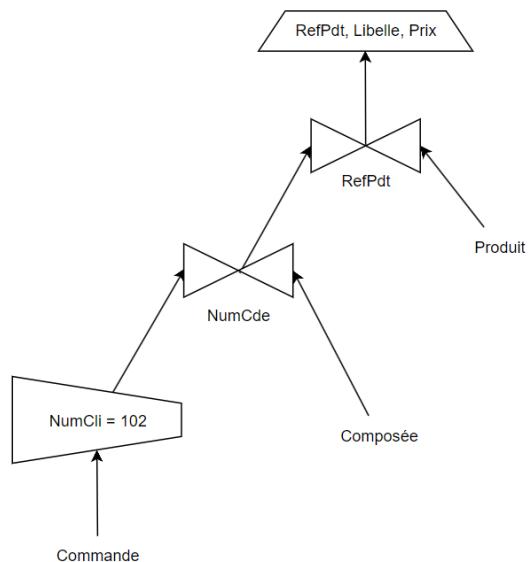
Version sans JOIN :

```
SELECT c.NumCli, c.Nom
FROM Client c, Commande co
WHERE c.NumCli = co.NumCli
AND co.Date = '2020-11-10';
```

2.5 Afficher les produits achetés par le client 102

2.5.1 Algèbre Relationnelle

$R_1 = \sigma_{NumCli = 102} (Commande)$
 $R_2 = (R_1)NumCde \bowtie NumCde (Composé)$
 $R_3 = (R_2)RefPdt \bowtie RefPdt (Produit)$
 $R_4 = \pi_{RefPdt, Libelle, Prix} (R_3)$



2.5.2 SQL

```

SELECT p.RefPdt, p.Libelle
FROM Produit p
JOIN Composee cp ON p.RefPdt = cp.RefPdt
JOIN Commande c ON cp.NumCde = c.NumCde
WHERE c.NumCli = 102;
    
```

Version sans JOIN :

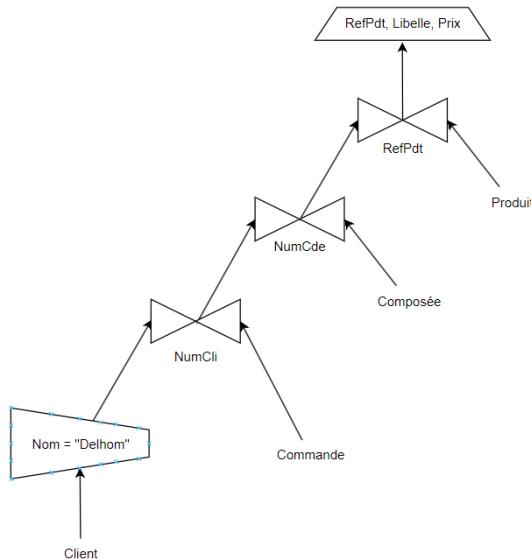
```

SELECT p.RefPdt, p.Libelle
FROM Produit p, Composee cp, Commande c
WHERE p.RefPdt = cp.RefPdt
    AND cp.NumCde = c.NumCde
    AND c.NumCli = 102;
    
```

2.6 Afficher les produits achetés par le client "Delhom"

2.6.1 Algèbre Relationnelle

$R_1 = \sigma_{Nom = "Delhom"}(Client)$
 $R_2 = (R_1)NumCli \bowtie NumCde (Commande)$
 $R_3 = (R_2) NumCde \bowtie NumPdt (Composé)$
 $R_4 = (R_3) RefPdt \bowtie Pdt (Produit)$.
 $R_5 = \pi_{RefPdt, Libelle, Prix} (R_4)$



2.6.2 SQL

```

SELECT p.RefPdt, p.Libelle
FROM Produit p
JOIN Composee cp ON p.RefPdt = cp.RefPdt
JOIN Commande c ON cp.NumCde = c.NumCde
JOIN Client cl ON c.NumCli = cl.NumCli
WHERE cl.Nom = 'Delhom';
  
```

Version sans JOIN :

```

SELECT p.RefPdt, p.Libelle
FROM Produit p, Composee cp, Commande c, Client cl
  
```

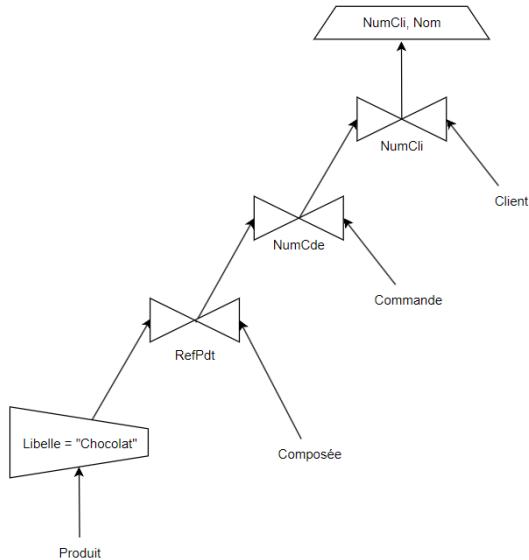
```

WHERE p . RefPdt = cp . RefPdt
    AND cp . NumCde = c . NumCde
    AND c . NumCli = cl . NumCli
    AND cl . Nom = 'Delhom';
  
```

2.7 Noms des clients ayant acheté du Chocolat

2.7.1 Algèbre Relationnelle

$R_1 = \sigma_{\text{Libelle} = \text{"Chocolat"}}(\text{Produit})$
 $R_2 = (R_1) \text{RefPdt} \bowtie \text{RefPdt}$ (Composé)
 $R_3 = (R_2) \text{NumCde} \bowtie \text{NumCde}$ (Commande)
 $R_4 = (R_3) \text{NumCli} \bowtie \text{NumCli}$ (Client)
 $R_5 = \pi_{\text{NumCli}, \text{Nom}}(R_4)$



2.7.2 SQL

```

SELECT cl . NumCli , cl . Nom
FROM Client cl
JOIN Commande c ON cl . NumCli = c . NumCli
  
```

```

JOIN Composee cp ON c.NumCde = cp.NumCde
JOIN Produit p ON cp.RefPdt = p.RefPdt
WHERE p.Libelle = 'Chocolat';

```

Version sans JOIN :

```

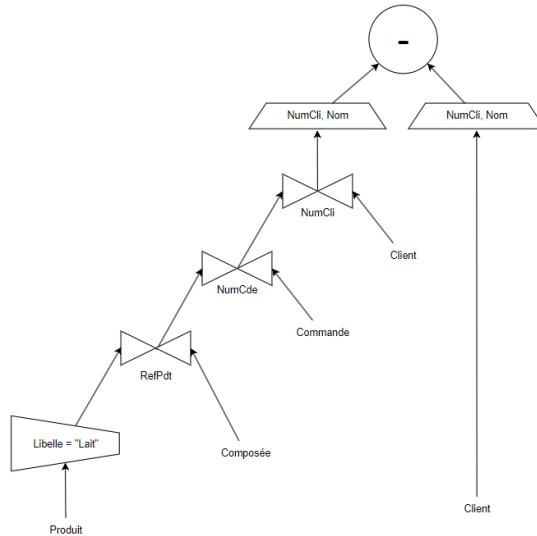
SELECT cl.NumCli, cl.Nom
FROM Client cl, Commande c, Composee cp, Produit p
WHERE cl.NumCli = c.NumCli
    AND c.NumCde = cp.NumCde
    AND cp.RefPdt = p.RefPdt
    AND p.Libelle = 'Chocolat';

```

2.8 Noms des clients n'ayant jamais acheté de lait

2.8.1 Algèbre Relationnelle

$R_1 = \sigma_{Libelle = "lait"}(Produit)$
 $R_2 = (R_1)RefPdt \bowtie RefPdt(Composee)$
 $R_3 = (R_2)NumCde \bowtie NumCde(Commande)$
 $R_4 = \pi_{NumCli, Nom}(R_3)$
 $R_5 = \pi_{NumCli, Nom}(Client)$
 $R_6 = R_5 - R_4.$



2.8.2 SQL

```
SELECT cl.NumCli, cl.Nom
FROM Client cl
WHERE cl.NumCli NOT IN (
    SELECT cl.NumCli
    FROM Commande c
    JOIN Composee cp ON c.NumCde = cp.NumCde
    JOIN Produit p ON cp.RefPdt = p.RefPdt
    WHERE cl.NumCli = c.NumCli AND p.Libelle = 'lait'
);
```

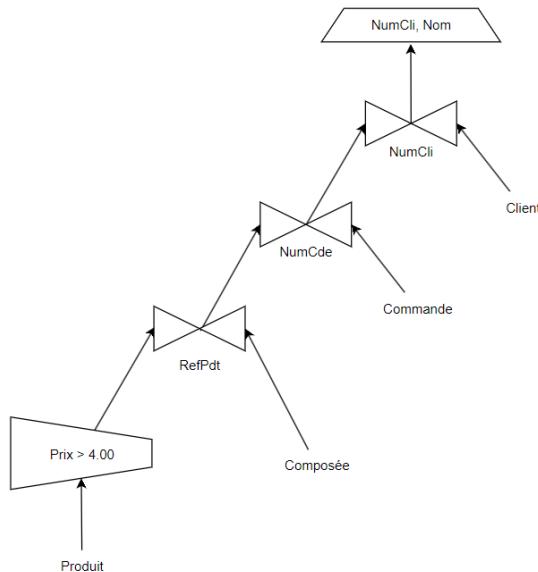
Version sans JOIN :

```
SELECT cl.NumCli, cl.Nom
FROM Client cl
WHERE cl.NumCli NOT IN (
    SELECT cl.NumCli
    FROM Commande c, Composee cp, Produit p
    WHERE c.NumCde = cp.NumCde
        AND cp.RefPdt = p.RefPdt
        AND cl.NumCli = c.NumCli
        AND p.Libelle = 'lait'
);
```

2.9 Noms des clients ayant acheté des produits dont le prix est supérieur à 4 Euros

2.9.1 Algèbre Relationnelle

$R_1 = \sigma_{\text{Prix} > 4,00\text{\euro}}(\text{Produit})$
 $R_2 = (R_1) \text{RefPdt} \bowtie \text{RefPdt} (\text{Compose}\acute{e})$
 $R_3 = (R_2) \text{NumCde} \bowtie \text{NumCde} (\text{Commande})$
 $R_4 = (R_3) \text{NumCli} \bowtie \text{NumCli} (\text{Client})$
 $R_5 = \pi_{\text{NumCli}, \text{Nom}}(R_4)$.



2.9.2 SQL

```

SELECT cl.NumCli cl.Nom
FROM Client cl
JOIN Commande c ON cl.NumCli = c.NumCli
JOIN Composee cp ON c.NumCde = cp.NumCde
JOIN Produit p ON cp.RefPdt = p.RefPdt
WHERE p.Prix > 4;
  
```

Version sans JOIN :

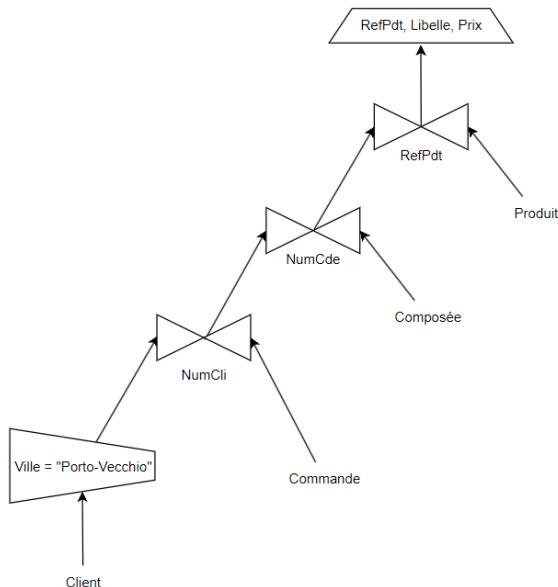
```

SELECT cl.NumCli cl.Nom
FROM Client cl, Commande c, Composee cp, Produit p
WHERE cl.NumCli = c.NumCli
    AND c.NumCde = cp.NumCde
    AND cp.RefPdt = p.RefPdt
    AND p.Prix > 4;
  
```

2.10 Liste des produits achetés par des porto-vecchiais

2.10.1 Algèbre Relationnelle

$R_1 = \sigma_{Ville = "Porto-Veccchio"}(Client)$
 $R_2 = (R_1) NumCli \bowtie NumCde$ (Commande)
 $R_3 = (R_2) NumCde \bowtie NomCde$ (Composée)
 $R_4 = (R_3) \Delta_{Nom\ RefPdt} \bowtie RefPdt$ (Produit)
 $R_5 = \pi_{RefPdt, Libelle, Prix}(R_4)$



2.10.2 SQL

```

SELECT DISTINCT p.Libelle
FROM Produit p
JOIN Composee cp ON p.RefPdt = cp.RefPdt
JOIN Commande c ON cp.NumCde = c.NumCde
JOIN Client cl ON c.NumCli = cl.NumCli
WHERE cl.Ville = 'Porto-Veccchio';
    
```

Version sans JOIN :

```
SELECT DISTINCT p.Libelle
FROM Produit p, Composee cp, Commande c, Client cl
WHERE p.RefPdt = cp.RefPdt
      AND cp.NumCde = c.NumCde
      AND c.NumCli = cl.NumCli
      AND cl.Ville = 'Porto-Vecchio';
```

3 Question 3

3.1 Création de la base de données

```

CREATE TABLE Client (
    NumCli INT PRIMARY KEY,
    Nom VARCHAR(50) NOT NULL,
    Adresse VARCHAR(100),
    CP CHAR(5),
    Ville VARCHAR(50),
    Tel VARCHAR(15)
);

CREATE TABLE Produit (
    RefPdt INT PRIMARY KEY,
    Libelle VARCHAR(100) NOT NULL,
    Prix DECIMAL(10, 2) NOT NULL
);

CREATE TABLE Commande (
    NumCde INT PRIMARY KEY,
    Date DATE NOT NULL,
    Montant DECIMAL(10, 2),
    NumCli INT NOT NULL,
    FOREIGN KEY (NumCli) REFERENCES Client(NumCli)
);

CREATE TABLE Composee (
    NumCde INT NOT NULL,
    RefPdt INT NOT NULL,
    Quantite INT NOT NULL,
    PRIMARY KEY (NumCde, RefPdt),
    FOREIGN KEY (NumCde) REFERENCES Commande(NumCde),
    FOREIGN KEY (RefPdt) REFERENCES Produit(RefPdt)
);

```

11. Total des commandes du client 102

J'ai interprété cette question de deux manières différentes : le **montant total** des commandes du client 102 et le **nombre total** de commandes du client 102.

3.1.1 Montant total des commandes du client 102

```

SELECT SUM(co.Montant) as "Montant-total-des-commandes-du-client-102"
FROM Commande co
WHERE co.NumCli = 102;

```

3.1.2 Nombre total de commandes du client 102

```
SELECT COUNT(co.NumCde) AS "Nombre total de commandes du client 102"  
FROM Commande co  
WHERE co.NumCli = 102;
```

12. Montant moyen des commandes

```
SELECT AVG(co.Montant) AS "Montant moyen des commandes"  
FROM Commande co;
```

13. Nombre des clients

```
SELECT COUNT(cl.NumCli) AS "Nombre des clients"  
FROM Client cl;
```