

1) Implémentons le modèle relationnel fourni en utilisant le SQL :

a) Création de la table client :

```
CREATE TABLE client (  
    id INT PRIMARY KEY,  
    Nom_client VARCHAR(15),  
    Tel_client INT );
```

b) Création de la table produit :

```
CREATE TABLE produit (  
    id_produit INT PRIMARY KEY,  
    Nom_produit VARCHAR(20),  
    Categories VARCHAR(20),  
    Price INT  
);
```

c) Création de la table commande :

```
CREATE TABLE commande (  
    id INT,  
    id_produit INT,  
    Date_commande DATE,  
    Quantite INT,  
    Montant_total INT,  
    FOREIGN KEY (id) REFERENCES client (id),  
    FOREIGN KEY (id_produit) REFERENCES produit (id_produit)  
);
```

2) Insérons des données dans les tables :

Pour la table client :

```
INSERT INTO client VALUES
```

(1, 'Ousmane', 784749451),  
(2, 'Abou', 771259787),  
(3, 'Amadou', 77777777),  
(4, 'Samba', 778889564),  
(5, 'Demba', 762555407),  
(6, 'Fanta', 759078312),  
(7, 'Kadia', 779087654);

Pour la table produit :

**INSERT INTO produit VALUES**

(1, 'Téléphone portable', 'Informatique', 8),  
(2, 'Ordinateur', 'Informatique', 1),  
(3, 'Parfum', 'Cosmétique', 1),  
(4, 'Costume', 'Vêtements', 2),  
(5, 'Paquet de lait', 'Aliments', 3),  
(6, 'Pelle', 'Outils', 5);

Pour la table commande :

**INSERT INTO commande VALUES**

(1, 6, '2023-12-01', 3000),  
(2, 5, '2024-11-10', 3000),  
(3, 5, '2023-12-12', 9000),  
(4, 3, '2024-02-12', 12000),  
(1, 2, '2023-05-19', 200000),  
(5, 4, '2023-03-19', 90000);

3) Écrivons une requête SQL pour récupérer les noms des clients qui ont passé une commande d'au moins un widget et d'au moins un gadget, ainsi que le coût total des widgets et gadgets commandés par chaque client.

**SELECT c.Nom\_client AS nom\_client,**  
**IFNULL(w.cout\_total\_widgets, 0) AS cout\_total\_widgets,**

```

IFNULL(g.cout_total_gadgets, 0) AS cout_total_gadgets

FROM client c

LEFT JOIN ( SELECT cm.id, SUM(cm.Quantite * p.price) AS cout_total_widgets
FROM commande cm
INNER JOIN produit p ON cm.id_produit = p.id_produit
WHERE p.Nom_produit = 'widget'
GROUP BY cm.id) AS w ON c.id = w.id

LEFT JOIN (SELECT cm.id, SUM(cm.Quantite * p.Price) AS cout_total_gadgets
FROM commande cm
INNER JOIN produit p ON cm.id_produit = p.id_produit
WHERE p.Nom_produit = 'gadget'
GROUP BY cm.id) AS g ON c.id = g.id

WHERE w.cout_total_widgets IS NOT NULL AND g.cout_total_gadgets IS NOT NULL;

```

4) Les noms des clients qui ont commandé au moins un widget et le coût total des widgets par client :

```

SELECT c.Nom_client AS 'Nom des clients', SUM(cm.Quantite * p.Price) AS 'Cout total des widgets'
FROM client c
INNER JOIN commande cm ON cm.id = c.id
INNER JOIN produit p ON p.id_produit = cm.id_produit
WHERE p.Nom_produit = 'widget'
GROUP BY c.id;

```

5) Les noms des clients qui ont commandé au moins un gadget et le coût total des gadgets par client :

```

SELECT c.Nom_client AS 'Nom des clients', SUM(cm.Quantite * p.Price) AS 'Cout total des gadgets'
FROM client c
INNER JOIN commande cm ON cm.id = c.id
INNER JOIN produit p ON p.id_produit = cm.id_produit
WHERE p.Nom_produit = 'gadget' GROUP BY c.id;

```

6) Écrivons une requête pour récupérer les noms des clients qui ont passé une commande d'au moins un doohickey, ainsi que le coût total des doohickeys commandés par chaque client.

```
SELECT c.Nom_client, SUM(cm.Quantite * p.Price) AS 'Cout total des doohickey'  
FROM client c  
INNER JOIN commande cm ON cm.id = c.id  
INNER JOIN produit p ON p.id_produit = cm.id_produit  
WHERE p.Nom_produit = 'doohickey'  
GROUP BY c.id;
```

7) Écrivons une requête pour récupérer le nombre total de widgets et de gadgets commandés par chaque client, ainsi que le coût total des commandes.

```
SELECT c.Nom_client,  
SUM(CASE WHEN p.nom_produit = 'widget' THEN 1 ELSE 0 END) AS total_widgets,  
SUM(CASE WHEN p.nom_produit = 'gadget' THEN 1 ELSE 0 END) AS total_gadgets,  
SUM(cm.quantite * p.Price) AS cout_total_commandes  
FROM client c  
INNER JOIN commande cm ON c.id = cm.id  
INNER JOIN produit p ON cm.id_produit = p.id_produit  
GROUP BY c.id;
```

8) Écrivons une requête pour récupérer les noms des produits qui ont été commandés par au moins un client, ainsi que la quantité totale de chaque produit commandé.

```
SELECT p.Nom_produit,  
COUNT(cm.Quantite) AS 'Quantité totale de produit commandée'  
FROM produit p  
INNER JOIN commande cm ON cm.id_produit = p.id_produit  
GROUP BY p.id_produit;
```

9) Écrivons une requête pour récupérer les noms des clients qui ont passé le plus de commandes, ainsi que le nombre total de commandes passées par chaque client.

```

SELECT c.Nom_client,
COUNT(cm.quantite) AS 'nombre_total_commandes'
FROM client c
INNER JOIN commande cm ON c.id = cm.id
GROUP BY c.id
ORDER BY COUNT(cm.quantite) DESC LIMIT 2;

```

10) Écrivez une requête pour récupérer les noms des produits les plus commandés, ainsi que la quantité totale de chaque produit commandé.

```

SELECT p.Nom_produit,
SUM(cm.Quantite) AS nombre_total_commandes
FROM produit p
INNER JOIN commande cm ON p.id_produit = cm.id_produit
GROUP BY p.id_produit
ORDER BY COUNT(cm.Quantite) DESC LIMIT 2;

```

11) Écrivez une requête pour récupérer les noms des clients qui ont passé une commande chaque jour de la semaine, ainsi que le nombre total de commandes passées par chaque client.

```

SELECT c.Nom_client,
COUNT(DISTINCT DAYOFWEEK(cm.date_commande)) AS jours_avec_commandes
FROM client c
INNER JOIN commande cm ON c.id = cm.id
GROUP BY c.id
HAVING jours_avec_commandes = 7;

```