

```
CREATE DATABASE DQL;
```

```
USE DQL;
```

```
CREATE TABLE customer (  
    customer_id INT PRIMARY KEY,  
    customer_Name VARCHAR(30) NOT NULL,  
    customer_Tel VARCHAR(30) NOT NULL  
);
```

```
CREATE TABLE product (  
    product_id INT PRIMARY KEY,  
    product_name VARCHAR(30) NOT NULL,  
    catogory VARCHAR(30) NOT NULL,  
    Price VARCHAR(30) NOT NULL  
);
```

```
CREATE TABLE orders (  
    orders_id INT PRIMARY KEY ,  
    product_id INT not null ,  
    customer_id INT not null ,  
    quantity INT not null,  
    OrdersDate VARCHAR(30) NOT NULL,  
    total_amount INT not null,  
    FOREIGN KEY (customer_id) REFERENCES customer(customer_id),  
    FOREIGN KEY (product_id) REFERENCES product(product_id)  
);
```

```
--insérer des enregistrements
```

```
INSERT INTO customer VALUES (1, 'cellou', '77120');
```

```
INSERT INTO customer VALUES (2, 'dior', '77130' );
```

```
INSERT INTO customer VALUES (3, 'awa', '77187');
```

```
SELECT*FROM customer;
```

```
INSERT INTO product VALUES (1, 'widget', 'mobile', '500');
```

```
INSERT INTO product VALUES (2, 'gadget', 'electro', '700' );
```

```
INSERT INTO product VALUES (3, 'doohickey', 'sante', '800');
```

```
SELECT*FROM product;
```

```
INSERT INTO orders VALUES (1, 1, 2,10, '10-01-2024', 5000);
```

```
INSERT INTO orders VALUES (2, 2, 1,12, '15-01-2024', 7000);
```

```
INSERT INTO orders VALUES (3, 3, 3,8, '20-01-2024', 8000);
```

```
INSERT INTO orders VALUES (4, 2, 2,11, '30-01-2024', 8500);
```

```
SELECT*FROM orders;
```

```
-- requête pour récupérer les noms des clients qui ont commandé au moins un widget et au moins un gadget ➤
```

```
-- ainsi que le coût total des widgets et gadgets commandés par chaque client.
```

```
-- Le coût de chaque article doit être calculé en multipliant la quantité par le prix du produit. ➤
```

```
SELECT customer.customer_Name, SUM(orders.quantity * product.Price) AS 'coût total' ➤
```

```
FROM customer
```

```
INNER JOIN product ON customer.customer_id = product.product_id
INNER JOIN orders ON product.product_id = orders.product_id
WHERE product.product_name = 'widget' OR product.product_name = 'gadget'
GROUP BY customer.customer_Name;

-- 2 Écrivez une requête pour récupérer les noms des clients qui ont commandé
-- au moins un widget,
-- ainsi que le coût total des widgets commandés par chaque client.
SELECT customer.customer_Name, SUM(orders.quantity * product.Price) AS 'coût
total'
FROM customer
INNER JOIN orders ON customer.customer_id = orders.customer_id
INNER JOIN product ON orders.product_id = product.product_id
WHERE product.product_name = 'widget'
GROUP BY customer.customer_Name;

-- récupérer les noms des clients qui ont commandé au moins un gadget
-- ainsi que le coût total des gadget commandés par chaque client.
SELECT customer.customer_name, SUM(orders.quantity * product.price) AS 'coût
total'
FROM customer
INNER JOIN orders ON customer.customer_id = orders.customer_id
INNER JOIN product ON orders.product_id = product.product_id
WHERE product.product_name = 'gadget'
GROUP BY customer.customer_name;

-- récupérer les noms des clients qui ont commandé au moins un doohickey
-- ainsi que le coût total des doohickey commandés par chaque client.
SELECT customer.customer_name, SUM(orders.quantity * product.price) AS 'coût
total'
FROM customer
INNER JOIN orders ON customer.customer_id = orders.customer_id
INNER JOIN product ON orders.product_id = product.product_id
WHERE product.product_name = 'doohickey'
GROUP BY customer.customer_name;

-- récupérer le nombre total de widgets et de gadgets commandés par chaque
-- client
-- ainsi que le coût total des commandes.
SELECT customer.customer_name,
SUM(CASE WHEN product.product_name = 'widget' THEN orders.quantity ELSE
0 END) AS 'Nombre Total de Widgets',
SUM(CASE WHEN product.product_name = 'gadget' THEN orders.quantity ELSE
0 END) AS 'Nombre Total de Gadgets',
SUM(orders.quantity * product.price) AS 'Coût Total des Commandes'
FROM customer
INNER JOIN orders ON customer.customer_id = orders.customer_id
INNER JOIN product ON orders.product_id = product.product_id
WHERE product.product_name IN ('widget', 'gadget')
GROUP BY customer.customer_name;

-- Écrivez 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 product.product_name,  
       SUM(orders.quantity) AS 'Quantité Totale'  
FROM customer  
INNER JOIN orders ON customer.customer_id = orders.customer_id  
INNER JOIN product ON orders.product_id = product.product_id  
GROUP BY product.product_name;
```

-- 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 cl.Customer_name, COUNT(o.orders_id) AS 'Nombre total de commandes'  
FROM Customer cl  
JOIN Orders o ON cl.Customer_id = o.Customer_id  
GROUP BY cl.Customer_name  
ORDER BY COUNT(o.orders_id) DESC;
```

-- requête pour récupérer les noms des produits les plus commandés

-- ainsi que la quantité totale de chaque produit commandé.

```
SELECT Product_name, sum(quantity) 'les noms des produits les plus commandés'  
FROM Customer cl  
JOIN Product p  
ON cl.Customer_id = p.Product_id  
JOIN Orders o  
ON p.Product_id = o.Product_id  
GROUP BY Product_name;
```

-- 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 cl.Customer_name, SUM(o.quantity) AS 'le nombre total de commandes  
       passées par chaque client'  
FROM Customer cl  
JOIN Orders o ON cl.Customer_id = o.Customer_id  
JOIN Product p ON o.Product_id = p.Product_id  
WHERE (MONTH(o.ordersDate) = 1 OR MONTH(o.ordersDate) = 7) AND YEAR  
       (o.ordersDate) = 2024  
GROUP BY cl.Customer_name;
```