**Votre Objectif**

À l’aide des tables fournies (Products, Customers, Orders, OrderDetails et ProductTypes), l’objectif est de formuler des requêtes SQL pour récupérer des données pertinentes à l’aide du langage DQL (Data Query Language).

**Tableau des produits :**

* **Enregistrement 1** : (ProductID= 1, ProductName= « Widget A », ProductType= « Widget », Price= 10.00)
* **Enregistrement 2** : (ProductID= 2, ProductName= « Widget B », ProductType= « Widget », Price= 15.00)
* **Enregistrement 3** : (ProductID= 3, ProductName= « Gadget X », ProductType= « Gadget », Price= 20.00)
* **Enregistrement 4** : (ProductID= 4, ProductName= « gadget Y », productType= « gadget », price= 25.00)
* **Enregistrement 5** : (ProductID= 5, ProductName= « Doohickey Z », ProductType= « Doohickey », Price= 30.00)

**Tableau des clients :**

* **Enregistrement 1 :** (CustomerID= 1, CustomerName= « John Smith », email= « john@example.com », Phone= « 123-456-7890 »)
* **Enregistrement 2 :** (CustomerID= 2, CustomerName= « Jane Doe », email= « jane.doe@example.com », Phone= « 987-654-3210 »)
* **Enregistrement 3 :** (CustomerID= 3, CustomerName= « Alice Brown », email= « alice.brown@example.com », Phone= « 456-789-0123 »)

**Tableau des commandes :**

* **Enregistrement 1 :** (OrderID= 101, CustomerID= 1, OrderDate= « 2024-05-01 »)
* **Enregistrement 2 :** (OrderID= 102, CustomerID= 2, OrderDate= « 2024-05-02 »)
* **Enregistrement 3** : (OrderID= 103, CustomerID= 3, OrderDate= « 2024-05-01 »)

**Tableau OrderDetails :**

* **Enregistrement 1 :** (OrderDetailID= 1, OrderID= 101, ProductID= 1, Quantity= 2)
* **Enregistrement 2 :** (OrderDetailID= 2, OrderID= 101, ProductID= 3, Quantity= 1)
* **Enregistrement 3 :** (OrderDetailID= 3, OrderID= 102, ProductID= 2, Quantity= 3)
* **Enregistrement 4 :** (OrderDetailID= 4, OrderID= 102, ProductID= 4, Quantity= 2)
* **Enregistrement 5 :** (OrderDetailID= 5, OrderID= 103, ProductID= 5, Quantity= 1)

**Tableau des types de produits :**

* **Enregistrement 1** : (ProductTypeID= 1, ProductTypeName= « Widget »)
* **Enregistrement 2** : (ProductTypeID= 2, ProductTypeName= « gadget »)
* **Enregistrement 3** : (ProductTypeID= 3, ProductTypeName= « Doohickey »)

**Instructions**

1. Récupérez tous les produits.
2. Récupérer tous les clients.
3. Récupérez toutes les commandes.
4. Récupérez tous les détails de la commande.
5. Récupérez tous les types de produits .
6. Récupérez les noms des produits qui ont été commandés par au moins un client, ainsi que la quantité totale de chaque produit commandé.
7. Récupérez 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.
8. Récupérez les noms des clients qui ont passé le plus de commandes, ainsi que le nombre total de commandes passées par chaque client.
9. Récupérez les noms des produits qui ont été le plus commandés, ainsi que la quantité totale de chaque produit commandé.
10. Récupérez les noms des clients qui ont passé une commande pour au moins un widget.
11. Récupérez les noms des clients qui ont passé une commande pour au moins un widget et au moins un gadget, ainsi que le coût total des widgets et gadgets commandés par chaque client.
12. Récupérez les noms des clients qui ont passé une commande pour au moins un gadget, ainsi que le coût total des gadgets commandés par chaque client.
13. Récupérez les noms des clients qui ont passé une commande pour au moins un doohickey, ainsi que le coût total des doohickeys commandés par chaque client.
14. Récupérez les noms des clients qui ont passé une commande tous les jours de la semaine, ainsi que le nombre total de commandes passées par chaque client.
15. Récupérez le nombre total de widgets et de gadgets commandés par chaque client, ainsi que le coût total des commandes.

----------------------- Point de contrôle de la gestion de la base de données relationnelle --------------------------

CREATE TABLE CodeLisible (

id INTEGER PRIMARY KEY,

Space TEXT NOT NULL

);

INSERT INTO CodeLisible VALUES (1, '--------------------Code lisible------------------');

--------- 1) Convertissez le diagramme de relation d'entité en modèle relationnel.

-- Create table Customer

CREATE TABLE Customer

(

Customer\_id INT PRIMARY KEY NOT NULL,

Customer\_name VARCHAR(100),

Customer\_Tel VARCHAR(50)

);

-- Create table Product

CREATE TABLE Product

(

Product\_id INT PRIMARY KEY NOT NULL,

Product\_name VARCHAR(150),

category VARCHAR(100),

price INT

);

-- Create table Orders

CREATE TABLE Orders

(

Customer\_id INT NOT NULL,

Product\_id INT NOT NULL,

orderDate DATE,

quantity INT,

total\_amount INT,

CONSTRAINT FK\_Customer FOREIGN KEY (Customer\_id) REFERENCES Customer(Customer\_id),

CONSTRAINT FK\_Product FOREIGN KEY (Product\_id) REFERENCES Product(Product\_id)

);

-- insert some Table Customer

INSERT INTO Customer VALUES (1, 'Justin', '77 203 54 53');

INSERT INTO Customer VALUES (2, 'Emmanuel', '76 891 25 95');

INSERT INTO Customer VALUES (3, 'Patrick', '70 321 45 78');

INSERT INTO Customer VALUES (4, 'Vanel Felixi', '78 693 85 24');

-- insert some Table Product

INSERT INTO Product VALUES (1, 'widget','Electronique',1500);

INSERT INTO Product VALUES (2, 'gadget', 'Appareil', 700);

INSERT INTO Product VALUES (3, 'doohickey', 'Tools', 2000);

INSERT INTO Product VALUES (4, 'widget', 'Electronique', 2500);

INSERT INTO Product VALUES (5, 'gadget', 'Appareil', 3000);

INSERT INTO Product VALUES (6, 'gadget', 'Tools', 150000);

-- insert some Table Orders

INSERT INTO Orders VALUES (4, 2, '2023-05-26', 3, 20000);

INSERT INTO Orders VALUES (3, 4, '2023-05-15', 6, 15000);

INSERT INTO Orders VALUES (2, 6, '2023-02-03', 7, 1050000);

INSERT INTO Orders VALUES (1, 5, '2023-02-05', 3, 9000);

INSERT INTO Orders VALUES (3, 3, '2023-05-08', 2, 4000);

INSERT INTO Orders VALUES (4, 1, '2023-05-19', 10, 15000);

INSERT INTO Orders VALUES (2, 1, '2023-05-06', 1, 2000);

INSERT INTO Orders VALUES (1, 2, '2023-05-18', 5, 3500);

<https://www.mycompiler.io/view/JMpSAnSzdaO> site interressant

-- 1) Écrivez une requête SQL 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 Space FROM CodeLisible;

SELECT Customer\_name, sum(quantity\*price) 'coût total'

FROM Customer cl

JOIN Product p

ON cl.Customer\_id = p.Product\_id

JOIN Orders o

ON p.Product\_id = o.Product\_id

Where p.Product\_name = 'widget' AND p.Product\_name = 'gadget';

SELECT Space FROM CodeLisible;

-- 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\_name, sum(quantity\*price) 'coût total'

FROM Customer cl

JOIN Product p

ON cl.Customer\_id = p.Product\_id

JOIN Orders o

ON p.Product\_id = o.Product\_id

Where p.Product\_name = 'widget';

SELECT Space FROM CodeLisible;

-- 3) Rédigez une requête pour récupérer les noms des clients qui ont passé une commande pour au moins un gadget,

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

SELECT Customer\_name, sum(quantity\*price) 'coût total'

FROM Customer cl

JOIN Product p

ON cl.Customer\_id = p.Product\_id

JOIN Orders o

ON p.Product\_id = o.Product\_id

Where p.Product\_name = 'gadget';

SELECT Space FROM CodeLisible;

-- 4) Écrivez une requête pour récupérer les noms des clients qui ont commandé au moins un doohickey,

-- ainsi que le coût total des doohickeys commandés par chaque client.

SELECT Customer\_name, sum(quantity\*price) 'coût total'

FROM Customer cl

JOIN Product p

ON cl.Customer\_id = p.Product\_id

JOIN Orders o

ON p.Product\_id = o.Product\_id

Where p.Product\_name = 'doohickey';

SELECT Space FROM CodeLisible;

-- 5) Rédigez une requête pour récupérer le nombre total de widgets et de gadgets commandés par chaque client,

-- ainsi que le coût total des commandes.

SELECT Customer\_name, sum(quantity) 'Nombre Total', sum(quantity\*price) 'coût total des commandes'

FROM Customer cl

JOIN Product p

ON cl.Customer\_id = p.Product\_id

JOIN Orders o

ON p.Product\_id = o.Product\_id

Where p.Product\_name = 'widget'AND p.Product\_name = 'gadget';

SELECT Space FROM CodeLisible;

-- 6) 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 Customer\_name, Product\_name, sum(quantity) 'la quantité totale'

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

having quantity >= 1;

SELECT Space FROM CodeLisible;

-- 7) Rédigez 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 Customer\_name, max(total\_amount) 'le Montant total de commande'

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

having max(total\_amount);

SELECT Space FROM CodeLisible;

-- 8) Rédigez une 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;

SELECT Space FROM CodeLisible;

-- 9) 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 Customer\_name, sum(quantity) 'le nombre total de commandes passées par chaque client'

FROM Customer cl

JOIN Product p

ON cl.Customer\_id = p.Product\_id

JOIN Orders o

ON p.Product\_id = o.Product\_id

WHERE orderDate LIKE '2023-02-%' OR orderDate LIKE '2023-05-%'

GROUP BY Customer\_name;