KPI POUR LA FINANCE

# **Les délais de paiement, encours & impayés**

## 

WITH ca\_payed AS(

SELECT SUM(amount) AS payed\_par\_annee, YEAR(paymentDate) AS annee\_payment

FROM payments

GROUP BY annee\_payment

),

ca\_commande AS (

SELECT SUM(quantityOrdered \* priceEach) AS commandes\_par\_annee, YEAR(orderDate) AS annee\_commande

FROM orderdetails od

JOIN orders ON orders.orderNumber = od.orderNumber

WHERE status = 'Shipped' - - pour ne prendre en compte que les commandes livrées

GROUP BY annee\_commande

)

SELECT

ca\_commande.annee\_commande AS annee2,

commandes\_par\_annee,

payed\_par\_annee,

ca\_commande.commandes\_par\_annee - ca\_payed.payed\_par\_annee AS reste\_a\_payer,

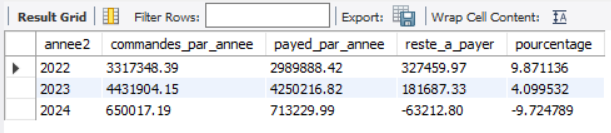
(ca\_commande.commandes\_par\_annee - ca\_payed.payed\_par\_annee)/ca\_commande.commandes\_par\_annee\*100 AS pourcentage

FROM ca\_commande

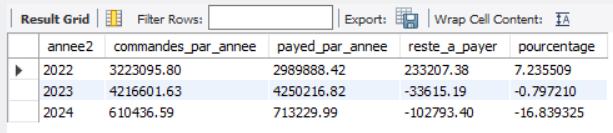
JOIN ca\_payed ON ca\_payed.annee\_payment = ca\_commande.annee\_commande

ORDER BY annee2;

**Résultat obtenu 1 :**



**Résultat en tenant compte des commandes non livrées**

****

**Remarques :**

La situation des encours s’est nettement améliorée entre 2022 et 2023.

Le résultat de 2024 n’est pas représentatif

# **Exploration des risques d’impayés : tableau complet par client**

-- tableau complet par client

WITH ca\_payed AS(

SELECT customerNumber,SUM(amount) AS paid\_par\_client, MAX(paymentDate) AS dernier\_paiement

FROM payments

GROUP BY customerNumber

),

ca\_commande AS (

SELECT od.orderNumber, SUM(quantityOrdered \* priceEach) AS ca\_par\_commande

FROM orderdetails od

JOIN orders ON orders.orderNumber = od.orderNumber

WHERE status = 'Shipped'

GROUP BY od.orderNumber

)

SELECT

c.customerName,

c.customerNumber,

SUM(ca\_commande.ca\_par\_commande) AS ca\_par\_client,

SUM(CASE

WHEN YEAR(orderDate)=2022 THEN ca\_commande.ca\_par\_commande

ELSE 0

END) AS CA\_par\_client\_2022,

SUM(CASE

WHEN YEAR(orderDate)=2023 THEN ca\_commande.ca\_par\_commande

ELSE 0

END) AS CA\_par\_client\_2023,

SUM(CASE

WHEN YEAR(orderDate)=2024 THEN ca\_commande.ca\_par\_commande

ELSE 0

END) AS CA\_par\_client\_2024,

ca\_payed.paid\_par\_client,

SUM(ca\_commande.ca\_par\_commande)-ca\_payed.paid\_par\_client AS reste\_a\_payer,

(SUM(ca\_commande.ca\_par\_commande)-ca\_payed.paid\_par\_client)\*100/(SUM(ca\_commande.ca\_par\_commande)) AS pourcentage,

ca\_payed.dernier\_paiement,

CASE WHEN (SUM(ca\_commande.ca\_par\_commande)-ca\_payed.paid\_par\_client) > 0 AND SUM(CASE

WHEN YEAR(orderDate)=2024 THEN ca\_commande.ca\_par\_commande

ELSE 0

END)=0 AND MONTH(dernier\_paiement) <11 THEN 'Rouge'

WHEN (SUM(ca\_commande.ca\_par\_commande)-ca\_payed.paid\_par\_client) > 0 AND SUM(CASE

WHEN YEAR(orderDate)=2024 THEN ca\_commande.ca\_par\_commande

ELSE 0

END) =0 AND MONTH(dernier\_paiement) =12 THEN 'Orange'

ELSE 0

END AS Alerte,

creditLimit,

ROUND((100\*(SUM(ca\_commande.ca\_par\_commande)-ca\_payed.paid\_par\_client)/creditLimit),2) AS jauge\_credit

FROM customers AS c

JOIN orders ON c.customerNumber = orders.customerNumber

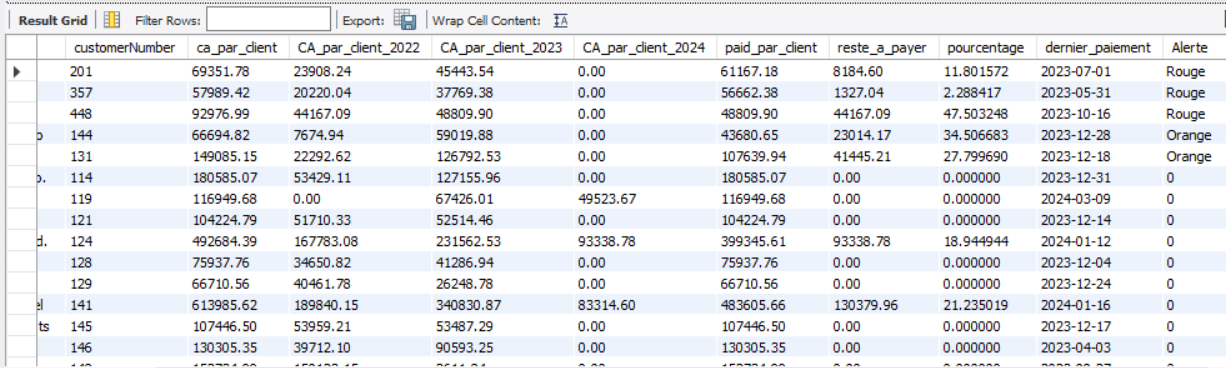
JOIN ca\_commande ON ca\_commande.orderNumber = orders.orderNumber

JOIN ca\_payed ON ca\_payed.customerNumber = c.customerNumber

GROUP BY c.customerName, c.customerNumber

ORDER BY reste\_a\_payer DESC;

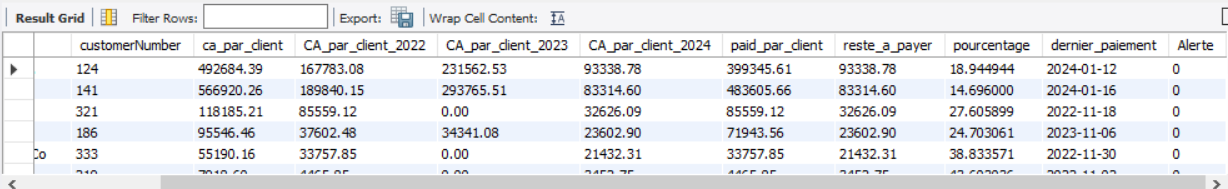
**Résultat obtenu :**



**Remarques:**

Donner l’alerte rouge et orange pour 5 clients

**Résultat obtenu en tenant compte des commandes non livrées**



Pas de problèmes

Etablir une jauge de crédit limite pour chaque client

# **Les marges**

Calcul des marges par produit

-- tableau des marges moyennes par produit

WITH ca\_produit AS(

SELECT productCode,

SUM(quantityOrdered\*priceEach) AS CA\_par\_produit,

SUM(quantityOrdered) AS quantités\_vendues\_par\_produit

FROM orderdetails

GROUP BY productCode

)

SELECT ca\_produit.productCode AS code\_produit,

productName AS nom\_du\_produit,

ca\_produit.CA\_par\_produit,

ca\_produit.quantités\_vendues\_par\_produit,

ca\_produit.CA\_par\_produit/ca\_produit.quantités\_vendues\_par\_produit AS prix\_de\_vente\_moyen\_par\_produit,

buyPrice,

(ca\_produit.CA\_par\_produit/ca\_produit.quantités\_vendues\_par\_produit)-buyPrice AS marge\_moyenne\_par\_produit,

ROUND(100\*(((ca\_produit.CA\_par\_produit/ca\_produit.quantités\_vendues\_par\_produit)-buyPrice)/buyPrice),2) AS taux\_marge\_moyenne,

MSRP,

ROUND(MSRP-(ca\_produit.CA\_par\_produit/ca\_produit.quantités\_vendues\_par\_produit),2) AS remises,

ROUND(100\*((MSRP-(ca\_produit.CA\_par\_produit/ca\_produit.quantités\_vendues\_par\_produit))/MSRP),2) AS taux\_remises\_MSRP

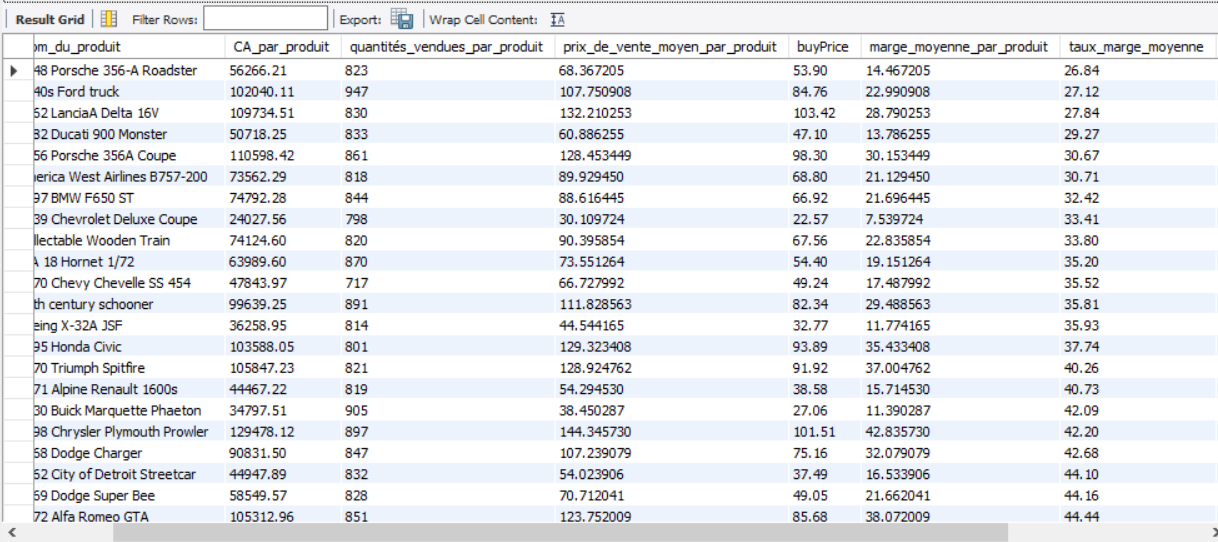
FROM products as p

JOIN ca\_produit ON ca\_produit.productCode = p.productCode

GROUP BY ca\_produit.productCode,ca\_produit.CA\_par\_produit,p.buyPrice

ORDER BY taux\_marge\_moyenne ASC

;



Résultat :

Le taux de marge moyen est calculé par rapport au prix d’achat.

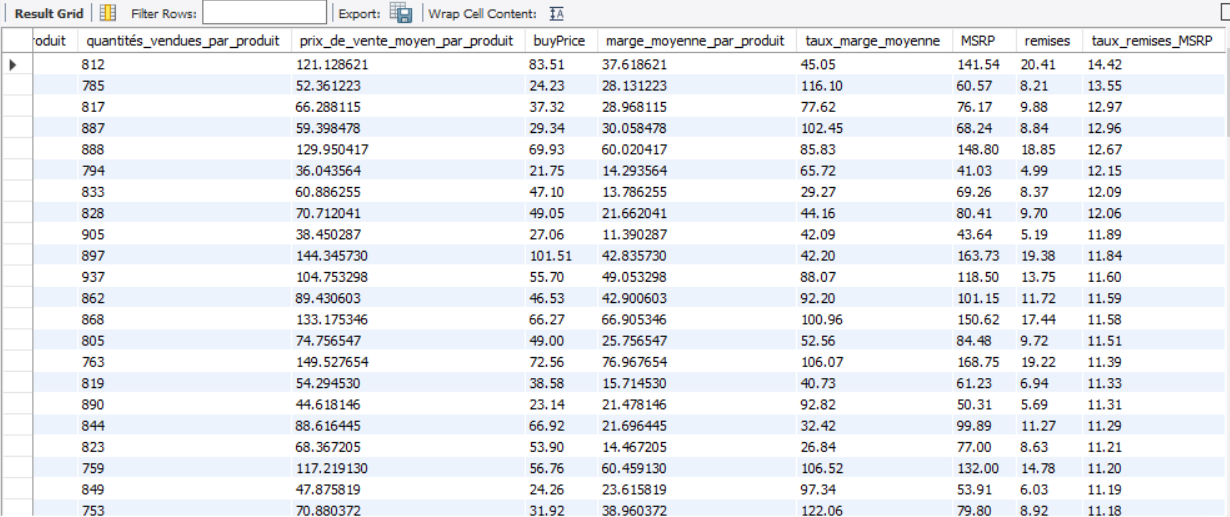
Quand ce taux dépasse 100%, cela signifie que le prix est plus que doublé à la vente

en revanche, les marges situées entre 26% et 75% sont trop basses

Le taux MSRP est calculé par rapport au MSRP

Résultat : classement décroissant du taux de remise/MSRP

Le plus élevé est de 14%



Calcul des marges globales - évolution dans le temps

WITH ca\_produit AS (

SELECT

YEAR(orderDate) AS annee,

orderdetails.productCode,

SUM(quantityOrdered \* priceEach) AS CA,

SUM(quantityOrdered) AS quantites\_vendues

FROM orderdetails

JOIN orders ON orderdetails.orderNumber = orders.orderNumber

GROUP BY annee, orderdetails.productCode

)

SELECT ca\_produit.annee,

SUM(ca\_produit.CA) AS CA,

SUM(ca\_produit.quantites\_vendues) AS quantites\_vendues,

SUM(ca\_produit.quantites\_vendues \* p.buyPrice) AS total\_achats,

SUM(ca\_produit.CA) - SUM(ca\_produit.quantites\_vendues \* p.buyPrice) AS marge\_globale,

(SUM(ca\_produit.CA) - SUM(ca\_produit.quantites\_vendues \* p.buyPrice))/SUM(ca\_produit.quantites\_vendues)AS marge\_globale\_moyenne,

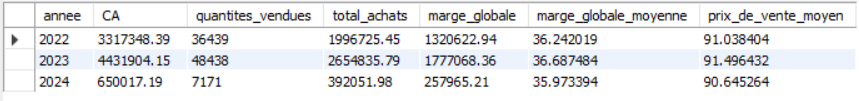
SUM(ca\_produit.CA)/SUM(ca\_produit.quantites\_vendues) AS prix\_de\_vente\_moyen

FROM products AS p

JOIN ca\_produit ON ca\_produit.productCode = p.productCode

GROUP BY ca\_produit.annee;

Résultat :



La marge suit le chiffres d’affaires, en moyenne (CA divisé par la quantité d’unités achetées) reste stable d’une année sur l’autre.

# **Les stocks**

Les stocks pèsent-ils globalement trop lourdement sur la trésorerie.

On compte en unités sans les prix, pour comparer le niveau d’activité

-- quantités en stock

SELECT SUM(quantityinstock)

from products;

**Résultat : 55517 unités**

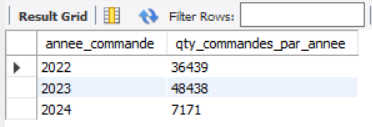
On compare avec le nombre d’unités vendues chaque année

SELECT YEAR(orderDate) AS annee\_commande, SUM(quantityOrdered) AS qty\_commandes\_par\_annee

FROM orderdetails od

JOIN orders ON orders.orderNumber = od.orderNumber

GROUP BY annee\_commande;



La quantité de stocks reflète une vision trop optimiste de l’activité

SELECT SUM(quantityInStock \* buyPrice) AS stock

FROM products;€

**Résultat:**

**Coût du stock : 3 053 685 €**

**Cela représente une immobilisation financière trop importante par rapport à l’activité**