Version 2:

**Requête des KPI**

KPI 1 : Le chiffre d’affaire

Le chiffre d’affaire total / variable temporelle

SELECT sum(priceEach \* quantityOrdered) as Chiffre\_affaire FROM orderdetails

JOIN orders ON orders.ordernumber= orderdetails.ordernumber;

Le chiffre d’affaire / produits → Ajouter variable temporelle

SELECT date\_format(orderdate, "%M %Y") as commande\_mois\_annee, productline, sum(quantityOrdered) as totalquantity, sum(orderdetails.priceeach \* quantityOrdered) as CA

FROM orderdetails

JOIN products ON products.productCode = orderdetails.productCode

JOIN orders ON orders.ordernumber= orderdetails.ordernumber

GROUP BY commande\_mois\_annee, productline;

KPI 2 : Les produits

Les plus vendus   
SELECT productCode, productName, productLine, quantityInStock, (AVG(priceEach) - buyPrice) as marge

FROM Orderdetails

Join products USING(productcode)

Group by productCode

ORDER BY marge DESC;  
Les moins vendus

WITH flop\_prod as (

SELECT productline, productname, sum(priceeach \* quantityordered) as ca\_total,

RANK() over (partition by productline order by sum(priceeach \* quantityordered) ) as rank\_prod

FROM products

LEFT JOIN orderdetails USING(productcode)

GROUP BY productline, productname

ORDER BY productline , CA\_total asc)

SELECT \* from flop\_prod

WHERE rank\_prod <= 5;

KPI 3 : Les clients

WITH ca\_evolution\_client as (

SELECT customerName ,sum(priceeach \* quantityordered) as ca\_total,

sum(CASE WHEN year(orderdate) = year(now()) THEN priceeach \* quantityordered

ELSE 0

END ) as ca\_N,

sum(CASE

WHEN year(orderdate) = year(now()) - 1 THEN priceeach \* quantityordered

ELSE 0

END ) as ca\_N\_1,

sum(CASE

WHEN year(orderdate) = year(now()) - 2 THEN priceeach \* quantityordered

ELSE 0

END ) as ca\_N\_2

FROM customers

LEFT JOIN orders USING(customernumber)

LEFT JOIN orderdetails USING(ordernumber)

WHERE year(orderdate) >= year(now()) - 2

GROUP BY customerName

ORDER BY customerName asc ),

ca\_total\_tous as (

SELECT sum(quantityordered \* priceeach) as ca\_tous\_temps

FROM orderdetails)

SELECT \* , (CA\_N - CA\_N\_1) \* 100 / CA\_N\_1 as RATIO\_24\_23, round(CA\_total \* 100/ ca\_tous\_temps,1) as percent\_ca\_tous\_temp,

CASE

WHEN CA\_N = 0 THEN "relancer le client"

WHEN CA\_N = 0 and CA\_N\_1 = 0 THEN "client perdu"

ELSE "autre"

END as classe\_client

FROM ca\_evolution\_client

CROSS JOIN ca\_total\_tous

ORDER BY percent\_ca\_tous\_temp desc ;

Autre methode :

SELECT customerName,

SUM(quantityordered \* priceeach) as CA,

SUM(quantityordered \* priceeach) \* 100

/

(SELECT SUM(od.quantityordered \* od.priceeach)

FROM orders

LEFT JOIN orderdetails as od USING(orderNumber)

WHERE `status` = "shipped" ) as total\_ca

FROM customers

LEFT JOIN orders USING(customerNumber)

LEFT JOIN orderdetails USING(orderNumber)

WHERE `status` = "shipped"

GROUP BY customerName

ORDER BY CA desc;

Premiere Version

Rébecca

**Requête des KPI**

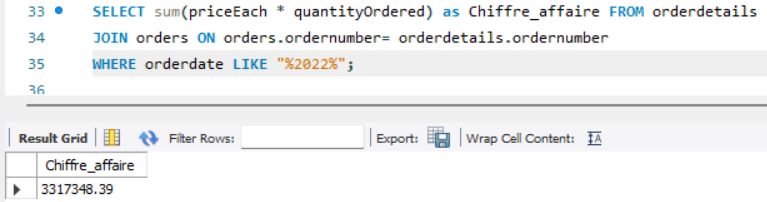
KPI 1 : CA total par année

-- Je fais le CA Total pour année 2022

SELECT sum(priceEach \* quantityOrdered) as Chiffre\_affaire FROM orderdetails

JOIN orders ON orders.ordernumber= orderdetails.ordernumber

WHERE orderdate LIKE "%2022%";

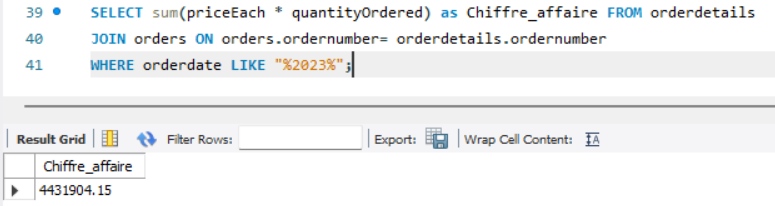


-- Je fais le CA Total pour année 2023

SELECT sum(priceEach \* quantityOrdered) as Chiffre\_affaire FROM orderdetails

JOIN orders ON orders.ordernumber= orderdetails.ordernumber

WHERE orderdate LIKE "%2023%";

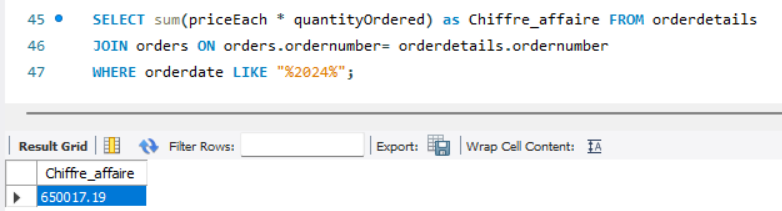


-- Je fais le CA Total pour année 2024 \*\*\*\*\* ONLY Q1 \*\*\*\*\*\*\*\*

SELECT sum(priceEach \* quantityOrdered) as Chiffre\_affaire FROM orderdetails

JOIN orders ON orders.ordernumber= orderdetails.ordernumber

WHERE orderdate LIKE "%2024%";



KPI 2 CA total par manager (RH)???

KPI 3 : Nombre de produits vendus par mois, par catégorie et par CA :

SELECT date\_format(orderdate, "%M %Y") as commande\_mois\_annee, productline, sum(quantityOrdered) as totalquantity, sum(orderdetails.priceeach \* quantityOrdered) as CA

FROM orderdetails

JOIN products ON products.productCode = orderdetails.productCode

JOIN orders ON orders.ordernumber= orderdetails.ordernumber

GROUP BY commande\_mois\_annee, productline;

Pour l’année 2024 uniquement :

SELECT date\_format(orderdate, "%M %Y") as commande\_mois\_annee, productline, sum(quantityOrdered) as totalquantity, sum(orderdetails.priceeach \* quantityOrdered) as CA

FROM orderdetails

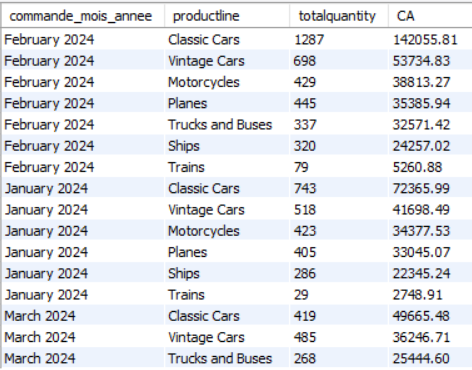
JOIN products ON products.productCode = orderdetails.productCode

JOIN orders ON orders.ordernumber= orderdetails.ordernumber

WHERE orderdate LIKE "%2024%"

GROUP BY commande\_mois\_annee, productline

ORDER BY commande\_mois\_annee, CA DESC;



Les catégories qui se vendent le plus sur l’année 2024 sont les “classic cars” et les “Vintage cars”.

KPI 2bis : Taux de variation N-1

utiliser wonder function

KPI 3: i want to see if we have customer that didnt do any order

SELECT customerName, country, orderdate FROM customers

LEFT JOIN orders ON orders.customerNumber = customers.customerNumber

WHERE orderdate is NULL

ORDER BY country;



KPI 4 : Les 10 produits les moins commandés, leur stock et leur MSRP

SELECT productcode, productName, productLine, quantityInStock, AVG(priceeach), MSRP, sum(quantityOrdered) as quantitevendue

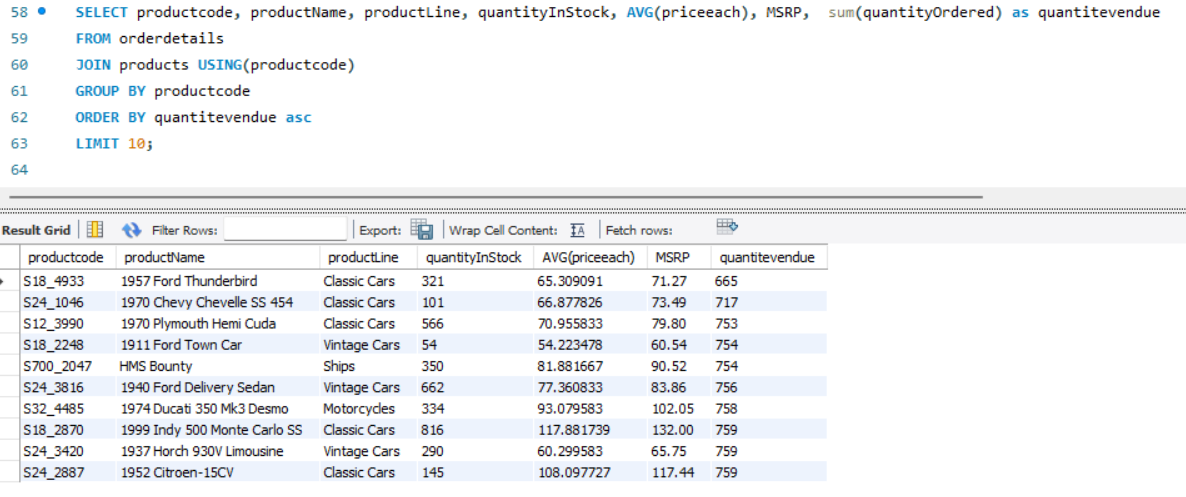
FROM orderdetails

JOIN products USING(productcode)

GROUP BY productcode

ORDER BY quantitevendue asc

LIMIT 10;



On est sous le MSRP pour tous les produits.

KPI 5 : Le produit qui fait le plus de ventes

SELECT productCode, productName, productLine, quantityInStock, (AVG(priceEach) - buyPrice) as marge

FROM Orderdetails

Join products USING(productcode)

Group by productCode

ORDER BY marge DESC;

