SQL QUERY

1.Total Revenue:

SELECT

CONCAT('$ ',SUM(price)) AS Total\_Revenue

FROM orderdetails;



2. Total Profit:

SELECT

CONCAT('$ ',SUM(o.price) - SUM(p.buyPrice\*o.quantityOrdered)) AS Total\_Profit

FROM products AS p

JOIN orderdetails AS o USING(productCode);



3. Total Quantity Ordered:

SELECT

SUM(quantityOrdered) AS Total\_Quantity\_Ordered

FROM orderdetails;



4. Total Product:

SELECT

COUNT(DISTINCT(productName)) AS Total\_Product

FROM products;



5. Total Customer:

SELECT

COUNT(DISTINCT(customerNumber)) AS Total\_Customer

FROM customers;



6. Revenue Generated / Product Line:

SELECT

productLine,

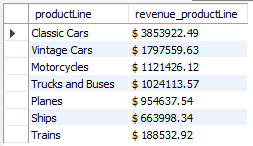
CONCAT('$ ',SUM(price)) AS revenue\_productLine

FROM products AS p

JOIN orderdetails AS o USING(productCode)

GROUP BY productLine

ORDER BY SUM(price) DESC;



7. Order Quantity / Product Line:

SELECT

productLine,

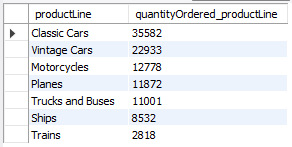
SUM(quantityOrdered) AS quantityOrdered\_productLine

FROM products AS p

JOIN orderdetails AS o USING(productCode)

GROUP BY productLine

ORDER BY SUM(quantityOrdered) DESC;



8. No of Customer / Product Line:

SELECT

productLine,

COUNT(DISTINCT(c.customerNumber)) AS countCustomer\_productLine

FROM customers AS c

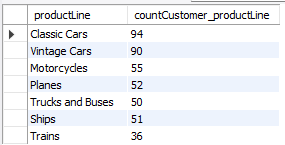
JOIN orders AS o USING(customerNumber)

JOIN orderdetails AS od USING(orderNumber)

JOIN products as p USING(productCode)

GROUP BY productLine

ORDER BY COUNT(c.customerNumber) DESC;



9. Top 5 Customer:

SELECT

customerName,

CONCAT('$ ',SUM(price)) AS top\_5\_customer

FROM customers AS c

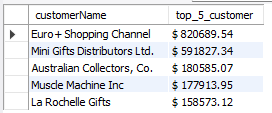
JOIN orders AS o USING(customerNumber)

JOIN orderdetails AS od USING(orderNumber)

GROUP BY customerName

ORDER BY SUM(price) DESC

LIMIT 5;



10. Bottom 5 Customer:

SELECT

customerName,

CONCAT('$ ',SUM(price)) AS bottomCustomer

FROM customers AS c

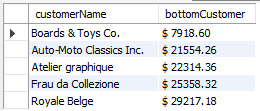
JOIN orders AS o USING(customerNumber)

JOIN orderdetails AS od USING(orderNumber)

GROUP BY customerName

ORDER BY SUM(price)

LIMIT 5;



11. Order Quantity Status:

SELECT

status,

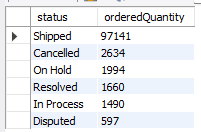
SUM(quantityOrdered) AS orderedQuantity

FROM orders AS o

JOIN orderdetails AS od USING(orderNumber)

GROUP BY status

ORDER BY SUM(quantityOrdered) DESC;



12. Delivered Orders:

SELECT

COUNT(DISTINCT(orderNumber)) AS Delivered\_Orders

FROM orders

WHERE status IN ('Disputed','In Process','Shipped');



13. Undelivered Orders:

SELECT

COUNT(DISTINCT(orderNumber)) AS Undelivered\_Orders

FROM orders

WHERE status IN ('On Hold','Resolved','Cancelled');



14. Total Orders: Count of orders:

SELECT

COUNT(DISTINCT(orderNumber)) AS Total\_Orders

FROM orders;



15. Revenue Generated / Month

SELECT

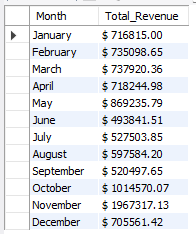
MONTHNAME(o.orderDate) AS Month,

CONCAT('$ ',SUM(od.price)) AS Total\_Revenue

FROM orders AS o

JOIN orderdetails AS od USING(orderNumber)

GROUP BY MONTHNAME(o.orderDate);



16. Profit Generated / Month:

SELECT

MONTHNAME(o.orderDate) AS Month,

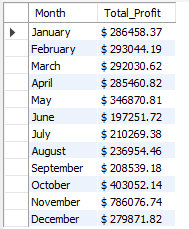
CONCAT('$ ',SUM(od.price) - SUM(p.buyPrice\*od.quantityOrdered)) AS Total\_Profit

FROM orders AS o

JOIN orderdetails AS od USING(orderNumber)

JOIN products AS p USING(productCode)

GROUP BY MONTHNAME(o.orderDate);



17. Percentage of profit / territory

CREATE VIEW Total\_profit AS

SELECT

SUM(od.price) - SUM(p.buyPrice\*od.quantityOrdered) AS Total\_profit

FROM orders AS o

JOIN orderdetails AS od USING(orderNumber)

JOIN products AS p USING(productCode);

CREATE VIEW Profit\_Territory AS

SELECT

office.territory,

SUM(od.price) - SUM(p.buyPrice\*od.quantityOrdered) AS Profit

FROM orders AS o

JOIN orderdetails AS od USING(orderNumber)

JOIN products AS p USING(productCode)

JOIN customers AS c USING(customerNumber)

JOIN employees AS e ON e.employeeNumber = c.salesRepEmployeeNumber

JOIN offices AS office USING(officeCode)

GROUP BY office.territory;

SELECT

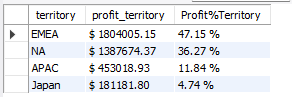
pt.territory,

CONCAT('$ ',pt.Profit) AS profit\_territory,

CONCAT(ROUND(pt.Profit / tp.Total\_Profit \* 100,2),' %') AS "Profit%Territory"

FROM Profit\_Territory AS pt

CROSS JOIN Total\_profit AS tp;



18. Year Trend based on Total\_revenue:

CREATE VIEW Revenue AS

SELECT

YEAR(o.orderDate) AS Year,

SUM(od.price) AS Revenue

FROM orders AS o

JOIN orderdetails AS od USING(orderNumber)

GROUP BY YEAR(o.orderDate);

CREATE VIEW Total\_Revenue AS

SELECT

SUM(od.price) AS "Total\_Revenue"

FROM orders AS o

JOIN orderdetails AS od USING(orderNumber);

SELECT

r.Year,

r.Revenue,

CONCAT(ROUND(r.Revenue / tr.Total\_Revenue \*100,2),' %') AS "Revenue vs Year"

FROM Revenue AS r

CROSS JOIN Total\_Revenue AS tr;

