\*\*JOINS:\*\*

1. Retrieve the `Product Name` and `Sales` for each product in the `Products` table, along with their corresponding `Order Quantity` and `Profit` from the `Orders` table.

SELECT P.ProductName, O.Sales, O.OrderQuantity, O.Profit

FROM Products P

INNER JOIN Orders O ON P.ProductID = O.ProductID;

1. Get a list of `Customer Name` and `Region` for customers who have placed orders, including those who haven't placed any orders yet.

SELECT C.CustomerName, C.Region

FROM Customers C

LEFT JOIN Orders O ON C.CustomerID = O.CustomerID;

1. List the `Product Name`, `Product Category`, and `Product Sub-Category` for products that have been ordered at least once, along with the total number of times they've been ordered.

SELECT P.ProductName, P.ProductCategory, P.ProductSubCategory, COUNT(O.ProductID) AS OrderCount

FROM Products P

LEFT JOIN Orders O ON P.ProductID = O.ProductID

GROUP BY P.ProductID, P.ProductName, P.ProductCategory, P.ProductSubCategory

HAVING OrderCount > 0;

1. Display the `Order ID`, `Customer Name`, and `Shipping Cost` for orders where the `Shipping Cost` is greater than the average shipping cost of all orders.

SELECT O.OrderID, C.CustomerName, O.ShippingCost

FROM Orders O

INNER JOIN Customers C ON O.CustomerID = C.CustomerID

WHERE O.ShippingCost > (SELECT AVG(ShippingCost) FROM Orders);

1. Retrieve the `Customer Name`, `Order Date`, and `Sales` for orders placed by customers in the 'West' region, ordered by `Sales` in descending order.

SELECT C.CustomerName, O.OrderDate, O.Sales

FROM Customers C

INNER JOIN Orders O ON C.CustomerID = O.CustomerID

WHERE C.Region = 'West'

ORDER BY O.Sales DESC;

1. Get a list of `Customer Name` and `Product Name` for customers who have placed orders, along with the corresponding product names, even if they haven't placed any orders.

SELECT C.CustomerName, IFNULL(P.ProductName, 'No Orders') AS ProductName

FROM Customers C

LEFT JOIN Orders O ON C.CustomerID = O.CustomerID

LEFT JOIN Products P ON O.ProductID = P.ProductID;

1. List the `Product Name` and `Product Category` for products that have been ordered by customers with the 'Corporate' customer segment.

SELECT P.ProductName, P.ProductCategory

FROM Products P

INNER JOIN Orders O ON P.ProductID = O.ProductID

INNER JOIN Customers C ON O.CustomerID = C.CustomerID

WHERE C.CustomerSegment = 'Corporate';

1. Retrieve the `Order ID`, `Product Name`, and `Discount` for orders that include products with a discount greater than 0.1.

SELECT O.OrderID, P.ProductName, O.Discount

FROM Orders O

INNER JOIN Products P ON O.ProductID = P.ProductID

WHERE O.Discount > 0.1;

1. Display the `Product Name`, `Sales`, and `Profit` for products that have been ordered at least twice, along with the total sales and profit for each product.

SELECT P.ProductName, SUM(O.Sales) AS TotalSales, SUM(O.Profit) AS TotalProfit

FROM Products P

INNER JOIN Orders O ON P.ProductID = O.ProductID

GROUP BY P.ProductName

HAVING COUNT(O.OrderID) >= 2;

1. Retrieve the `Customer Name` and `Region` for customers who have placed the highest number of orders, ordered by the order count in descending order.

SELECT C.CustomerName, C.Region, COUNT(O.OrderID) AS OrderCount

FROM Customers C

LEFT JOIN Orders O ON C.CustomerID = O.CustomerID

GROUP BY C.CustomerID, C.CustomerName, C.Region

ORDER BY OrderCount DESC;

1. Find the `Customer Name` and `Region` for customers who have placed orders on products with a discount greater than 0.1.

SELECT C.CustomerName, C.Region

FROM Customers C

WHERE C.CustomerID IN (SELECT DISTINCT CustomerID FROM Orders WHERE Discount > 0.1);

1. List the `Product Name` and `Product Category` for products that have been ordered by customers from the 'East' region, using a subquery to find customer IDs.

SELECT P.ProductName, P.ProductCategory

FROM Products P

INNER JOIN Orders O ON P.ProductID = O.ProductID

WHERE O.CustomerID IN (SELECT CustomerID FROM Customers WHERE Region = 'East');

1. Display the `Order Date` and `Sales` for orders that have a sales value greater than the average sales value of all orders.

SELECT OrderDate, Sales

FROM Orders

WHERE Sales > (SELECT AVG(Sales) FROM Orders);

1. Get a list of `Product Name` and `Product Category` for products that have been ordered more than the average number of times.

SELECT P.ProductName, P.ProductCategory

FROM Products P

WHERE P.ProductID IN (SELECT ProductID FROM Orders GROUP BY ProductID HAVING COUNT(\*) > (SELECT AVG(OrderCount) FROM (SELECT ProductID, COUNT(\*) AS OrderCount FROM Orders GROUP BY ProductID) AS AvgOrderCount));

1. Retrieve the `Customer Name` and `Customer Segment` for customers who have placed orders with a total `Profit` greater than the average profit of all orders.

SELECT C.CustomerName, C.CustomerSegment

FROM Customers C

WHERE C.CustomerID IN (SELECT CustomerID FROM Orders GROUP BY CustomerID HAVING SUM(Profit) > (SELECT AVG(Profit) FROM Orders));

1. List the `Product Name` and `Product Category` for products that have been ordered by customers in the 'Central' region, using a subquery to filter orders by region.

SELECT P.ProductName, P.ProductCategory

FROM Products P

INNER JOIN Orders O ON P.ProductID = O.ProductID

WHERE O.CustomerID IN (SELECT CustomerID FROM Customers WHERE Region = 'Central');

1. Display the `Order ID` and `Shipping Cost` for orders that have a shipping cost greater than the average shipping cost of orders placed by customers with the 'Consumer' segment.

SELECT O.OrderID, O.ShippingCost

FROM Orders O

INNER JOIN Customers C ON O.CustomerID = C.CustomerID

WHERE C.CustomerSegment = 'Consumer' AND O.ShippingCost > (SELECT AVG(ShippingCost) FROM Orders);

1. Get a list of `Customer Name` and `Region` for customers who have placed orders on products that belong to the 'Office Supplies' category.

SELECT C.CustomerName, C.Region

FROM Customers C

INNER JOIN Orders O ON C.CustomerID = O.CustomerID

INNER JOIN Products P ON O.ProductID = P.ProductID

WHERE P.ProductCategory = 'Office Supplies';

1. Retrieve the `Order ID` and `Order Date` for orders that have a quantity greater than the average order quantity of all orders.

SELECT O.OrderID, O.OrderDate

FROM Orders O

WHERE O.OrderQuantity > (SELECT AVG(OrderQuantity) FROM Orders);

1. Retrieve the `Product Name` and `Sales` for products that have been ordered by customers who have also placed orders in the 'West' region.

SELECT P.ProductName, O.Sales

FROM Products P

INNER JOIN Orders O ON P.ProductID = O.ProductID

WHERE O.CustomerID IN (SELECT CustomerID FROM Customers WHERE Region = 'West');