**Retrieve customer name, orderid, and order date for customers who have placed orders.**

SELECT

customer.customername,

order\_t.orderid,

order\_t.orderdate

FROM

order\_t

JOIN

customer

ON

order\_t.customerid = customer.customerid;

Alternative way:

SELECT

customer.customername,

order\_t.orderid,

order\_t.orderdate

FROM

order\_t,

customer

WHERE

order\_t.customerid = customer.customerid;

A screenshot of a computer

Description automatically generated

**For each line item of an order, list the orderid, order date, description of the product that was sold, amount charged (quantity multiplied by productprice), customerid and name of the customer who placed the order. Give amount charged the alias of Amount.**

SELECT

order\_t.orderid,

order\_t.orderdate,

product.productdescription,

orderline.quantity \* product.productprice AS Amount,

customer.customerid,

customer.customername

FROM

product,

order\_t,

customer,

orderline

WHERE

customer.customerid = order\_t.customerid

AND order\_t.orderid = orderline.orderid

AND orderline.productid = product.productid;

A screenshot of a computer

Description automatically generated

**For each order, list the orderid, the total amount and the total units on that order. Give the total amount the alias of TotalAmount, and the total units the alias of TotalUnits.**

Q3

SELECT

orderline.orderid,

SUM(orderline.quantity \* product.productprice) AS TotalAmount,

SUM(orderline.quantity) AS TotalUnits

FROM

orderline,

product

WHERE

orderline.productid = product.productid

GROUP BY

orderline.orderid;

Alternative way:

SELECT

orderline.orderid,

SUM(orderline.quantity \* product.productprice) AS TotalAmount,

SUM(orderline.quantity) AS TotalUnits

FROM

orderline

JOIN

product

ON

orderline.productid = product.productid

GROUP BY

orderline.orderid;

A screenshot of a computer

Description automatically generated

**Consider each group contains the products that have the same product description with the same product finish. For each group, list the product description, product finish, and the total units (sum of quantity) ordered. Give the total units the alias of TotalUnits.**

SELECT

product.productdescription,

product.productfinish,

SUM(orderline.quantity) AS TotalUnits

FROM

orderline

JOIN

product

ON

product.productid = orderline.productid

GROUP BY

product.productdescription,

product.productfinish;

Alternative way:

SELECT

product.productdescription,

product.productfinish,

SUM(orderline.quantity) AS TotalUnits

FROM

orderline,

product

WHERE

product.productid = orderline.productid

GROUP BY

product.productdescription,

product.productfinish;

A table with black text

Description automatically generated

**Which product lines have (include) two or more products? Your results should include the product line name and the number of products included in the product line. Also give the number of products the alias of NoofProducts.**

SELECT

productline.productlinename,

COUNT(product.productlineid) AS NoofProducts

FROM

product

JOIN

productline

ON

product.productlineid = productline.productlineid

GROUP BY

productline.productlinename

HAVING

COUNT(product.productlineid) >= 2;

Alternative way:

SELECT

productline.productlinename,

COUNT(product.productlineid) AS NoofProducts

FROM

product,

productline

WHERE

product.productlineid = productline.productlineid

GROUP BY

productline.productlinename

HAVING

COUNT(product.productlineid) >= 2;

A white background with black text

Description automatically generated

**Retrieve customer ID, customer name, and order date, and that will display customer data even if the customer has not placed an order.**

SELECT

customer.customerid,

customer.customername,

order\_t.orderdate

FROM

customer

LEFT OUTER JOIN

order\_t

ON

order\_t.customerid = customer.customerid;

A table of information

Description automatically generated with medium confidence

**Retrieve the productid, product description, and product price for each product whose price is greater than the average price of all products.**

SELECT

productid,

productdescription,

productprice

FROM

product

WHERE

productprice >

(SELECT

AVG(productprice)

FROM

product);

A screenshot of a computer

Description automatically generated

L**ist the customer name for orders that were placed before 2022-11-09.**

SELECT

customer.customername

FROM

customer

WHERE

customer.customerid IN

(SELECT

order\_t.customerid

FROM order\_t

WHERE orderdate < '2022-11-09');

A screenshot of a computer

Description automatically generated