**Case Study: SQL 2**

Please copy your **(1) SQL statement (as text), and (2) the result table (as image, screen prints of your output/results from DB Browser for SQLite)** from each SQL statement below:

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

SELECT customername,orderid,orderdate

FROM customer c, order\_t ot

WHERE c.customerid=ot.customerid;

Table

Description automatically generated

1. For each line item of an order, list the orderid, order date, description of the product that was sold, amount charged, customerid and name of the customer who placed the order.

SELECT ol.orderid,ot.orderdate,p.productdescription,(ol.quantity\*p.productprice) as Alias,c.customerid,c.customername

FROM customer c, product p, orderline ol, order\_t ot

WHERE ol.orderid=ot.orderid AND ol.productid=p.productid AND ot.customerid=c.customerid;

Table

Description automatically generated

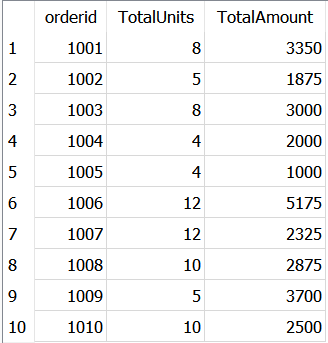
1. For each order, list the orderid, the total amount and the total units on that order.

SELECT ol.orderid,sum(ol.quantity) AS TotalUnits, sum(ol.quantity\*p.productprice) AS TotalAmount

FROM orderline ol,product p

WHERE ol.productid=p.productid

GROUP by ol.orderid;



1. 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 ordered.

SELECT p.productdescription,p.productfinish,sum(ol.quantity) AS TotalUnits

FROM orderline ol,product p

WHERE ol.productid=p.productid

GROUP by p.productdescription,p.productfinish;

Table

Description automatically generated

1. Which product lines have (include) two or more products?

SELECT pl.productlinename,count(p.productlineid) AS NoofProducts

FROM product p, productline pl

WHERE pl.productlineid=p.productlineid

GROUP BY pl.productlinename

HAVING COUNT(p.productlineid)>=2;

Table

Description automatically generated

1. Write a query that will retrieve customer ID, customer name, and order date, and that will display customer data even if the customer has not placed an order.

SELECT c.customerid,c.customername,ot.orderdate

FROM customer c

LEFT OUTER JOIN order\_t ot on c.customerid=ot.customerid;

****

1. Write a query that will 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);

Table

Description automatically generated

1. Now write a query that will list the customer name for orders that were placed before 2022-11-09.

SELECT customername

FROM customer

WHERE customerid IN(

SELECT customerid

FROM order\_t

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

Graphical user interface, text, application, chat or text message

Description automatically generated