1. ***Explain why it would be preferable to use a DATE data type to store date data instead of a character data type. (RQ1)***

Ans: If the date data is stored in character data type, the system will not be able to figure out its numeric value. Also, it becomes impossible for us to do numerical operations on the date data like add and subtract. By using DATE data type, we can do numerical operation and transform date formats by using functions provided by the system.

1. ***Rewrite the following WHERE clause without the use of the IN special operator: WHERE V\_STATE IN (‘TN’, ‘FL’, ‘GA’) (RQ7)***

Ans: WHERE V\_STATE = ‘TN’ AND V\_STATE = ‘FL’ AND V\_STATE = ‘GA’

1. ***Explain why the following two commands produce different results: (RQ9)***

***SELECT DISTINCT COUNT (V\_CODE) FROM PRODUCT;***

***SELECT COUNT (DISTINCT V\_CODE) FROM PRODUCT;***

Ans: DISTINCT COUNT(V\_CODE) will count the number of values in V\_CODE including **duplicated** values and give then it **will** eliminate any duplicated count numbers. COUNT(DISTINCT V\_CODE) applies the DISTINCT first so it will count the number of **distinct** values in V\_CODE and **will** **not** eliminate any duplicated count numbers after that.

1. ***In a SELECT query, what is the difference between a WHERE clause and a HAVING clause? (RQ11)***

Ans: HAVING clause is used after a GROUP BY command to filter the result of summarized data. WHERE clause should be used before GROUP BY to filter the rows of the table before they are grouped. Hence, you cannot use aggregate functions in WHERE clause.

1. ***Write a query to count the number of invoices. (P9)***

SELECT COUNT(\*) AS NUMBER\_OF\_INVOICES FROM INVOICE;

图形用户界面, 文本, 应用程序

中度可信度描述已自动生成

1. ***Write a query to count the number of customers with a balance of more than $500. (P10)***

SELECT COUNT(\*) AS NUMBER\_OF\_CUSTOMERS

FROM CUSTOMER

WHERE CUS\_BALANCE > 500;

图片包含 图形用户界面

描述已自动生成

1. ***Using the output shown in Figure P7.12 as your guide, generate a list of customer purchases, including the subtotals for each of the invoice line numbers. The subtotal is a derived attribute calculated by multiplying LINE\_UNITS by LINE\_PRICE. Sort the output by customer code, invoice number, and product description. Be certain to use the column aliases as shown in the figure. (P12)***

SELECT A.CUS\_CODE,

B.INV\_NUMBER,

D.P\_DESCRIPT,

ROUND(C.LINE\_UNIT,0) AS 'Units Bought',

C.LINE\_PRICE AS 'Unit Price',

ROUND(C.LINE\_UNIT\*C.LINE\_PRICE,2) AS Subtotal

FROM CUSTOMER AS A

JOIN INVOICE AS B ON A.CUS\_CODE = B.CUS\_CODE

JOIN LINE AS C ON B.INV\_NUMBER = C.INV\_NUMBER

JOIN PRODUCT AS D ON C.P\_CODE = D.P\_CODE;

表格

描述已自动生成

1. ***Use a query to compute the total of all purchases, the number of purchases, and the average purchase amount made by each customer. Your output values must match those shown in Figure P7.15. Sort the results by customer code. (P15)***

SELECT A.CUS\_CODE,

A.CUS\_BALANCE,

ROUND(SUM(C.LINE\_UNIT\*C.LINE\_PRICE),2) AS 'Total Purchases',

COUNT(\*) AS 'Number of Purchases',

ROUND(SUM(C.LINE\_UNIT\*C.LINE\_PRICE)/COUNT(\*),2) AS 'Average Purchases Amount'

FROM CUSTOMER AS A

JOIN INVOICE AS B ON A.CUS\_CODE = B.CUS\_CODE

JOIN LINE AS C ON B.INV\_NUMBER = C.INV\_NUMBER

JOIN PRODUCT AS D ON C.P\_CODE = D.P\_CODE

GROUP BY A.CUS\_CODE;

***表格

描述已自动生成***

1. ***Find the listing of customers who did not make purchases during the invoicing period. Sort the results by customer code. Your output must match the output shown in Figure P7.23. (P23)***

SELECT A.CUS\_CODE, A.CUS\_BALANCE

FROM CUSTOMER AS A

LEFT JOIN INVOICE AS B ON A.CUS\_CODE = B.CUS\_CODE

WHERE B.INV\_NUMBER IS NULL;

表格

描述已自动生成

1. ***Write a query to display the eight departments in the LGDEPARTMENT table sorted by department name. (P27)***

SELECT \*

FROM LGDEPARTMENT

ORDER BY DEPT\_NAME;

表格

描述已自动生成

1. ***Write a query to display the SKU (stock keeping unit), description, type, base, cat- egory, and price for all products that have a PROD\_BASE of Water and a PROD\_ CATEGORY of Sealer (Figure P7.28). (P28)***

SELECT PROD\_SKU, PROD\_DESCRIPT, PROD\_TYPE, PROD\_BASE, PROD\_CATEGORY, PROD\_PRICE

FROM LGPRODUCT

WHERE PROD\_BASE = 'Water' AND PROD\_CATEGORY = 'Sealer';

表格

描述已自动生成

1. ***Write a query to display the first name, last name, street ,city, state, and zip code of any customer who purchased a Foresters Best brand top coat between July 15, 2015, and July 31, 2015. If a customer purchased more than one such product, display the customer’s information only once in the output. Sort the output by state, last name, and then first name (Figure P7.32). (P32)***

SELECT DISTINCT CUST\_FNAME, CUST\_LNAME, CUST\_STREET, CUST\_CITY, CUST\_STATE, CUST\_ZIP

FROM LGCUSTOMER AS A

JOIN LGINVOICE AS B ON A.CUST\_CODE = B.CUST\_CODE

JOIN LGLINE AS C ON B.INV\_NUM = C.INV\_NUM

JOIN LGPRODUCT AS D ON C.PROD\_SKU = D.PROD\_SKU

JOIN LGBRAND AS E ON D.BRAND\_ID = E.BRAND\_ID

WHERE E.BRAND\_NAME = 'FORESTERS BEST' AND D.PROD\_CATEGORY = 'Top Coat' AND (B.INV\_DATE BETWEEN '2017-07-15' AND'2017-07-31')

ORDER BY CUST\_STATE, CUST\_LNAME, CUST\_FNAME;

报纸上的文字

中度可信度描述已自动生成