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.

2. ***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'

3. ***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.

4. ***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.

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

   SELECT COUNT(*) AS NUMBER_OF_INVOICES FROM INVOICE;

```
MariaDB [Ch07_SaleCo]> SELECT COUNT(*) AS NUMBER_OF_INVOICES
    -> FROM INVOICE;
+--------------------+
| NUMBER_OF_INVOICES |
+--------------------+
|                  8 |
+--------------------+
1 row in set (0.01 sec)
```

6. ***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;
```

```
MariaDB [Ch07_SaleCo]> SELECT COUNT(*) AS NUMBER_OF_CUSTOMERS
    -> FROM CUSTOMER
    -> WHERE CUS_BALANCE > 500;
+---------------------+
| NUMBER_OF_CUSTOMERS |
+---------------------+
|                   2 |
+---------------------+
1 row in set (0.01 sec)
```

7. ***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;
```

```
MariaDB [Ch07_SaleCo]> 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;
+----------+------------+-----------------------------------+--------------+------------+----------+
| CUS_CODE | INV_NUMBER | P_DESCRIPT                        | Units Bought | Unit Price | Subtotal |
+----------+------------+-----------------------------------+--------------+------------+----------+
|    10011 |       1002 | Rat-tail file, 1/8-in. fine       |            2 |       4.99 |     9.98 |
|    10011 |       1004 | Rat-tail file, 1/8-in. fine       |            3 |       4.99 |    14.97 |
|    10011 |       1004 | Claw hammer                       |            2 |       9.95 |    19.90 |
|    10011 |       1008 | PVC pipe, 3.5-in., 8-ft           |            5 |       5.87 |    29.35 |
|    10011 |       1008 | Steel matting, 4'x8'x1/6", .5" mesh |            3 |     119.95 |   359.85 |
|    10011 |       1008 | Claw hammer                       |            1 |       9.95 |     9.95 |
|    10012 |       1003 | B&D cordless drill, 1/2-in.       |            1 |      38.95 |    38.95 |
|    10012 |       1003 | Hrd. cloth, 1/4-in., 2x50         |            1 |      39.95 |    39.95 |
|    10012 |       1003 | 7.25-in. pwr. saw blade           |            5 |      14.99 |    74.95 |
|    10014 |       1001 | 7.25-in. pwr. saw blade           |            1 |      14.99 |    14.99 |
|    10014 |       1001 | Claw hammer                       |            1 |       9.95 |     9.95 |
|    10014 |       1006 | 1.25-in. metal screw, 25          |            3 |       6.99 |    20.97 |
|    10014 |       1006 | B&D jigsaw, 12-in. blade          |            1 |     109.92 |   109.92 |
|    10014 |       1006 | Claw hammer                       |            1 |       9.95 |     9.95 |
|    10014 |       1006 | Hicut chain saw, 16 in.           |            1 |     256.99 |   256.99 |
|    10015 |       1007 | 7.25-in. pwr. saw blade           |            2 |      14.99 |    29.98 |
|    10015 |       1007 | Rat-tail file, 1/8-in. fine       |            1 |       4.99 |     4.99 |
|    10018 |       1005 | PVC pipe, 3.5-in., 8-ft           |           12 |       5.87 |    70.44 |
+----------+------------+-----------------------------------+--------------+------------+----------+
18 rows in set (0.00 sec)
```

8. ***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;
```

```
MariaDB [Ch07_SaleCo]> 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;
+----------+-------------+----------------+--------------------+--------------------------+
| CUS_CODE | CUS_BALANCE | Total Purchases | Number of Purchases | Average Purchases Amount |
+----------+-------------+----------------+--------------------+--------------------------+
|    10011 |        0.00 |          444.00 |                  6 |                    74.00 |
|    10012 |      345.86 |          153.85 |                  3 |                    51.28 |
|    10014 |        0.00 |          422.77 |                  6 |                    70.46 |
|    10015 |        0.00 |           34.97 |                  2 |                    17.49 |
|    10018 |      216.55 |           70.44 |                  1 |                    70.44 |
+----------+-------------+----------------+--------------------+--------------------------+
5 rows in set (0.01 sec)
```

9. ***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;

```
MariaDB [Ch07_SaleCo]> 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;
+----------+-------------+
| CUS_CODE | CUS_BALANCE |
+----------+-------------+
|    10010 |        0.00 |
|    10013 |      536.75 |
|    10016 |      221.19 |
|    10017 |      768.93 |
|    10019 |        0.00 |
+----------+-------------+
5 rows in set (0.00 sec)
```

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

SELECT *

FROM LGDEPARTMENT

ORDER BY DEPT_NAME;

```
MariaDB [Ch07_LargeCo]> SELECT *
    -> FROM LGDEPARTMENT
    -> ORDER BY DEPT_NAME;
+----------+------------------+---------------+------------+---------+
| DEPT_NUM | DEPT_NAME        | DEPT_MAIL_BOX | DEPT_PHONE | EMP_NUM |
+----------+------------------+---------------+------------+---------+
|      600 | ACCOUNTING       | 957           | 555-2333   |   84583 |
|      250 | CUSTOMER SERVICE | 100           | 555-5555   |   84001 |
|      500 | DISTRIBUTION     | 348           | 555-3624   |   84052 |
|      280 | MARKETING        | 848           | 555-8500   |   84042 |
|      300 | PURCHASING       | 222           | 555-4873   |   83746 |
|      200 | SALES            | 475           | 555-2824   |   83509 |
|      550 | TRUCKING         | 842           | 555-0057   |   83683 |
|      400 | WAREHOUSE        | 789           | 555-1003   |   83759 |
+----------+------------------+---------------+------------+---------+
8 rows in set (0.01 sec)
```

11. ***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';

```
MariaDB [Ch07_LargeCo]> SELECT PROD_SKU, PROD_DESCRIPT, PROD_TYPE, PROD_BASE,  PROD_CATEGORY, PROD_PRICE
    -> FROM LGPRODUCT
    -> WHERE PROD_BASE = 'Water' AND PROD_CATEGORY = 'Sealer';
+----------+-----------------------------------------+-----------+-----------+---------------+------------+
| PROD_SKU | PROD_DESCRIPT                           | PROD_TYPE | PROD_BASE | PROD_CATEGORY | PROD_PRICE |
+----------+-----------------------------------------+-----------+-----------+---------------+------------+
| 1403-TUY | Sealer, Water Based, for Concrete Floors | Interior  | Water     | Sealer        |      42.99 |
+----------+-----------------------------------------+-----------+-----------+---------------+------------+
1 row in set (0.01 sec)
```

12. ***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;

```
MariaDB [Ch07_LargeCo]> 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;
+------------+------------+--------------------------+---------------+------------+----------+
| CUST_FNAME | CUST_LNAME | CUST_STREET              | CUST_CITY     | CUST_STATE | CUST_ZIP |
+------------+------------+--------------------------+---------------+------------+----------+
| LUPE       | SANTANA    | 1292 WEST 70TH PLACE     | Phenix City   | AL         | 36867    |
| HOLLIS     | STILES     | 1493 DOLLY MADISON CIRCLE | Snow Hill     | AL         | 36778    |
| LISETTE    | WHITTAKER  | 339 NORTHPARK DRIVE      | Montgomery    | AL         | 36197    |
| DEANDRE    | JAMISON    | 1571 HANES STREET        | Miami         | FL         | 33169    |
| CATHLEEN   | WHITMAN    | 1712 NORTHFIELD DRIVE    | Marshallville | GA         | 31057    |
| SHERIE     | STOVER     | 640 MOUNTAIN VIEW DRIVE  | Parksville    | KY         | 40464    |
| BRYCE      | HOGAN      | 1860 IMLACH DRIVE        | Newbury       | MA         | 01951    |
| SHELBY     | SALAS      | 486 SUSITNA VIEW COURT   | North Tisbury | MA         | 02568    |
| JERMAINE   | HANCOCK    | 1627 SAUNDERS ROAD       | Ellicott City | MD         | 21041    |
| WHITNEY    | WHITFIELD  | 1259 RHONE STREET        | Phippsburg    | ME         | 04567    |
| MONROE     | ALLISON    | 272 SCHODDE STREET       | Kalamazoo     | MI         | 49002    |
| DARLEEN    | PARRA      | 561 COLLIE HILL WAY      | Madison       | MS         | 39130    |
| CLINTON    | AGUIRRE    | 1651 VANGUARD DRIVE      | Franklinville | NC         | 27248    |
| TOMMIE     | PALMER     | 933 ELCADORE CIRCLE      | Arapahoe      | NC         | 28510    |
| JEFFEREY   | MCBRIDE    | 1043 ROCKRIDGE DRIVE     | Glenwood      | NJ         | 07418    |
| SIDNEY     | GARZA      | 772 SHEPPARD DRIVE       | Fair Harbor   | NY         | 11706    |
| TAMELA     | GUIDRY     | 1873 BAXTER ROAD         | Brooklyn      | NY         | 11252    |
| KAREN      | LEVINE     | 1534 PALMER COURT        | Cincinnati    | OH         | 45218    |
| STEPHENIE  | MCKENZIE   | 1039 DELAWARE PLACE      | Wilkes Barre  | PA         | 18763    |
| LAN        | NICHOLS    | 367 LAKEVIEW DRIVE       | Pittsburgh    | PA         | 15262    |
| KASEY      | SOSA       | 975 WEST 96TH AVENUE     | Kinzers       | PA         | 17535    |
| SHELBY     | THAYER     | 1634 RUANE ROAD          | Bordeaux      | SC         | 29835    |
| WILSON     | BELL       | 1127 CUNNINGHAM STREET   | Louisville    | TN         | 37777    |
| RENATE     | LADD       | 652 LEWIS STREET         | Crystal City  | VA         | 22202    |
| MELONIE    | JIMENEZ    | 848 DOWNEY FINCH LANE    | East Monkton  | VT         | 05443    |
+------------+------------+--------------------------+---------------+------------+----------+
25 rows in set (0.02 sec)
```