SQL - ASSIGNMENT

Customer

CNUM	CNAME	CITY	RATING	SNUM
2001	Hoffman	London	100	1001
2002	Glovanne	Rome	200	1003
2003	Liu	San Jose	300	1002
2004	Grass	Berlin	100	1002
2006	Clemens	London	300	1007
2007	Pereira	Rome	100	1004

Orders

ONUM	AMOUNT	ODATE	CNUM	SNUM
3001	18.69	3-oct-16	2003	1007
3003	767.19	3-oct-16	2001	1001
3002	1800.1	3-oct-16	2002	1004
3005	5160.45	3-oct-16	2003	1002
3006	1098.16	3-oct-16	2003	1007
3009	1713.23	4-oct-16	2002	1003
3007	75.75	4-oct-16	2004	1002
3008	4723	5-oct-16	2006	1001
3010	1309.95	6-oct-16	2004	1002
3011	8891.78	6-oct-16	2006	1001

Salespeople

SNUM	SNAME	CITY	COMM
1001	Peel	London	0.12
1002	Serres	San Jose	0.13
1004	Motika	London	0.11
1007	Rifkin	Barcelona	0.15
1003	Axelrod	New York	0.10

(Following Queries are based on above tables)

Qno.

1. Display snum, sname, city and comm. of all salespeople.
2. Display all snum without duplicates from all orders.
3. Display names and commission of all salespeople from London.
4. All Customers with a rating of 100.

5. Produce order no, amount and date for all rows in the order table.
6. All customers in San Jose, who have a rating > 200.
7. All customers who were either located in San Jose or had a rating above 200.
8. All order for more than 1000.
9. Names and cities of all salespeople in London with a commission above 0.10.
10. All customers excluding those with rating <= 100 unless they are located in Rome.
11. All Salespeople either in Barcelona or in London.
12. All salespeople with commissions between 0.10 and 0.12. (Boundary values 0.10 and 0.12 should be excluded.)
13. All customers with NULL values in city column.
14. ALL orders taken on Oct 3rd or 4th 1994. (Two ways)
15. All customers serviced by Peel or Motika.
16. All customers whose names begin with a letter from A to B.
17. All orders except those with 0 or NULL values in Amt field.
18. Count the number of salespeople currently listing orders in the order table.
19. Largest order taken by each salesperson, datewise.
20. Largest order taken by each salesperson with order value more than 5000.
21. Which day had the highest total amount ordered?
22. Count all order of 3rd Oct.
23. Count the number of different non-NULL city values in customers table.
24. Select each customers smallest order.
25. First customer in alphabetic order whose name begins with G.
26. Get the output like-For dd/mm/yy, there are ________ orders.
27. Assume that each salesperson has a 12% commission. Produce order no., salesperson no. and amount of salesperson's commission for that order.
28. Find highest rating in each city, put the output in this form-For the city (city), the highest rating is: (rating)
29. Display the totals of orders for each day and place the results in descending order.
30. All combinations of salespeople and customers who shared a city (i.e. same city).
31. Name of all customers matched with the salespeople serving them.

32. List each order number followed by the name of the customer who made the order.
33. Names of salesperson and customer for each order after the order number.
34. Produce all customers serviced by salespeople with a commission above 12%.
35. Calculate the amount of the salesperson's commission on each order with a rating above 100.
36. Find all pairs of customers having the same rating.
37. Find all pairs of customers having the same rating, each pair coming once only.
38. Policy is to assign three salesperson to each customer, one at each of the three ratings. Display all such combinations.
39. Display all customers located in cities where salesman Serres has customer.
40. Find all pair of customers served by a single salesperson.
41. Produce all pairs of salespeople who are living in the same city. Exclude combinations of salespeople with themselves as well as duplicates with the order reversed.
42. Produce all pair of orders by a given customer, names that customer and eliminates duplicates.
43. Produce names and titles of all customers with the same rating as Hoffman.
44. Extract all the orders of Motika.
45. All orders credited to the same salesperson who services Hoffman.
46. All order that are greater than the average for Oct 4th.
47. Find average commission of salespeople in London.
48. Find all orders attributed to salespeople in London.
49. Extract commissions of all salespeople servicing customers in London.
50. Find all customers whose cnum is 1000 above the snum of Serres.
51. Count the customers with rating above San Jose's average.
52. Obtain all orders for the customer named Cisnerous. (Assume you don't know his customer no. (cnum)).
53. Produce the names and rating of all customers who have above average orders.
54. Find total amount, in orders for each salesperson for whom this total is greater than the amount of the largest order in the table.
55. Find all customers with orders on Oct 3rd.

56. Find names and numbers of all salespeople who have more than one customer.
57. Check if correct salesperson was credited with each sale.
58. Find all orders with above average amounts for their customers.
59. Find the snums of the amounts from order table grouped by date, eliminating all those dates where the sum was not at least 2000 above the maximum amount.
60. Find names and numbers of all customers with ratings equal to the maximum for their city.
61. Find all salespeople who have customers in their cities who they don’t service. (Two solutions: one using join and other using correlated subquery.)
62. Extract cnum, cname and city from customer table if and only if one or more of the customers in the table are located in San Jose.
63. Find salespeople no who have multiple customer.
64. Find salespeople number, name and city who have multiple customers.
65. Find salespeople who serve only one customer.
66. Extract rows of all salespeople with more than one current order.
67. Find all salespeople who have customers with a rating of 300. (Use EXISTS)
68. Find all salespeople who have customers with a rating of 300. (Use JOIN)
69. Select all salespeople with customers located in their cities who are not assigned to them. (Use EXISTS)
70. Extract from customers table, every customer assigned to a salesperson who currently has at least one, other customer (besides the customer being selected) with orders in the order table.
71. Find salespeople with customers located in cities. (Two ways - one using ANY, other using IN)
72. Find all salespeople for whom there are customers that follow them in alphabetical order. (Two ways - one using ANY, other using EXISTS)
73. Select customers who have a greater rating than any customer in Rome.
74. Select all orders that had amounts that were greater than at least one of the orders from Oct 6th.

75. Find all orders with amounts smaller than any amount for a customer in San Jose. (Two ways - one using ANY, other without ANY)
76. Select those customers whose ratings are higher than every customer in Paris. (Two ways - one using ALL, other using NOT EXISTS)
77. Select all customers whose ratings are equal to or greater than ANY of the Serres.
78. Find all salespeople who have no customers located in their city. (Two ways - one using ANY, other using ALL)
79. Select all orders for amount greater than any for the customers in London.
80.
81. Find all salespeople and customers located in London.
82. For every salesperson, dates on which highest and lowest orders were brought.
83. List all of the salespeople and indicate those who don't have customers in their cities as well as those who do have.
84. Append strings to the selected fields, indicating whether or not a given salesperson was matched to a customer in his city.
85. Create a union of two queries that shows the names, cities and ratings of all customers. Those with a rating of 200 or greater will also have the word "High Rating", while the others will have the words "Low Rating".
86. Write a command that produces the name and number of each salesperson and each customer with more than one current order. Put the result in alphabetical order.
87. Form a union of three queries. Have the first select the snums of all salespeople in San Jose, the second cnums of all customers in San Jose and the third the onums of all orders on Oct 3rd. Retain duplicates between the last tow queries. But eliminate any redundancies between either of them and the first.
88. Insert a new row in customer table, having values London, Hoffman and 2001 of columns city, cname and cnum.
89. Assume that there is another table Londonstaff, having the same structure as salespeople. Insert all the rows from salespeople with city in London.
90. Assume another table daytotals having two attributes date and total. Insert rows from order table.

91. Delete all rows from the table salespeople.
92. Remove all orders from customers Clemens from the order table.
93. Increase the rating of all customers in Rome by 100.
94. Salesperson Serres has left the company. Assign her customers to Motika.
95. Assume a table sjpeople with column definitions that match those of salespeople who have customers in San Jose, whether the salespeople reside there or not.
96. Assume we have a table called samecity in which we store salespeople with customers in their home cities. Fill the table.
97. Assume we have a table - Bonus - containing snums of the salespeople as well as the dates and amounts of the maximum orders. Fill the table from existing order table.
98. London office has closed. Remove all customers assigned to salespeople in London.
99. Delete all salespersons who have at least one customer with a rating of 100 from salespeople table.
100. Find the lowest order for each day and delete the salespersons who produced it from salespeople table.
101. Same as above, but Peel should not be deleted.
102. Increase comm. by 0.01 of salespeople who have been assigned at least two customers.
103. Reduce the comm. by 0.01 of salespeople who have produced smallest orders.
104. Assume there is a table called Multicast, with all of the same column definitions as salespeople. Insert all salespeople with more than one customer into this table.
105. Delete all customers with no current orders.
106. Increase the commission by twenty percent of all salespeople with total current orders above 3000.
107. Write a command that will enable a user to pull order grouped by date out of the orders table quickly.
108. Create an index that would permit each salesperson to retrieve his/her orders grouped by date quickly.
109. Let us assume that each salesperson is to have only one customer of a given rating. Write a command to enforce it.
110. Create a view Londonstaff that consists of all salespeople from London.
111. Create a view that consists of various ratings and the counts.

112. Suppose each day we have to keep track of the number of salespeople taking orders, the number of orders, the average amount ordered and the total amount ordered. Create a corresponding view.
113. Create a view to see salespersons and customers by name for each order.
114. Create a view to see all of Axelrod’s orders and her commissions.
115. The company provides a bonus for the salesperson who has the customer with highest order on any given date. Create a view.
116. Same as above, but bonus will go only to salespeople when they had highest order at least ten times.
117. Create a view that shows all of the customers who have the highest rating.
118. Create a view that shows the number of salespeople in each city.
119. Create a view that shows the average and total orders for each salesperson after his/her name, (Assume that all names are unique.)
120. Create a view that shows each salesperson with multiple customers.
121. Create a view that gives counts, averages and tables for the orders on each order date.
122. Produce all the salespeople in London who had at least one customer located there as well. (Use intersect)
123. Produce all the salespeople in London who did not have customers there.
124. We want to see salespeople matched to their customers without excluding those salespeople who were not currently assigned to any customers. (Use outer join.)
125. Allow Janet to query the customers table, put restrict her access to those customers whose rating is the lowest.