Assignment 2- SQL Queries

Provide the required SQL query for each number.

Note: This assignment builds on your previous exercise where you have the salesman and customer table.

1. Using your create query, create the following table

   Order1(Orderno,customer_id, prod_id, date1, qty, salesman_id)

2. Enter the following data:

CUSTOMER_ID	PROD_ID	DATE1	QTY	SALESMAN_ID	ORDERNO
3006	104	10/15/2019	5	5002	9001
3006	108	10/15/2019	6	5002	9001
3009	109	10/20/2019	3	5007	9002
3002	102	10/14/2018	5	5001	9003
3002	101	10/14/2018	3	5001	9003

   Provide at least one insert query that enters a record in the Order1 table.

3. Choose an appropriate primary key for this table. Using the alter SQL statement, alter the table to apply a primary key constraint.

4. Using the alter SQL statement, add another column in the customer table name it as **referred.**

5. Using update query update the customer records as follows:

CUSTOMER_ID	CUST_NAME	CITY	GRADE	SALESMAN_ID	REFERRED
3006	Fabian Johnson	Paris	300	5006	-
3002	Nick Rimando	New York	100	5001	3006
3007	Brad Davis	New York	200	5002	-
3005	Graham Zusi	California	200	5002	-
3008	Julian Green	London	300	5002	3003
3009	Geoff Cameron	Berlin	100	5003	-
3003	Jozy Altridor	Moscom	200	5007	-

6. Retrieve all records in orders1 and sort it by quantity in a descending order.

7. Retrieve all records of customer 3006 in the order1 table.

8. Retrieve all the records of orders in the orders table based on date. The range is 10/14/208 to 10/15/2019. Hint: Use the TO_DATE function

9. Select all salesman with names that has “au” in them. Display all the names in upper case format.

10. Retrieve the salesman name that has orders in the order table. Remove repetitions in the results.

11. Show the customer names and the customer who referred them. Use an alias to differentiate between the two columns.

12. Using outer join, show all the customers with orders and without orders.

Output is similar to the one below:

CUST_NAME	ORDERNO
Fabian Johnson	9001
Fabian Johnson	9001
Geoff Cameron	9002
Nick Rimando	9003
Nick Rimando	9003
Graham Zusi	-
Jozy Altridor	-
Julian Green	-
Brad Davis	-

13. Using the UNION operator combine the cities between customer and salesman. The final list should not have repetitions.

14. Using the intersect operator, show only the common cities between salesman and customer.

15. Show the cities of customer without the cities that are common between salesman and customer. In other words, remove the cities that are common between salesman and customer
.
16. Sum the quantity of the order1 table by order number. Include the salesman name and the order number in the results.

17. Revise your number 15 by further filtering the results. Show only those records with a sum of quantity that is greater than 10.

18. Using a subquery, display all customers that has a grade greater than customer 3002.

19. Using a subquery in the select statement display the cust_name, grade and the overall average of the grade table.

Sample output.

CUST_NAME	GRADE	(SELECTAVG(GRADE)FROMCUSTOMER)
Fabian Johnson	300	200
Nick Rimando	100	200
Brad Davis	200	200
Graham Zusi	200	200

20. Retrieve all records of customer that has a grade greater than the grade of salesman who has a commission rate greater than .14. A subquery should be used.

Sample output:

3006	Fabian Johnson	Paris	300	5006
3007	Brad Davis	New York	200	5002
3005	Graham Zusi	California	200	5002
3008	Julian Green	London	300	5002
3003	Jozy Altridor	Moscow	200	5007

21. Group the customers by salesman ID and from this group get the maximum grade. Based from this query retrieve customers whose grade are in the grade of your first query. Hint: You have to use multiple column.

Sample output:

3006	Fabian Johnson	Paris	300	5006	-
3002	Nick Rimando	New York	100	5001	3006
3008	Julian Green	London	300	5002	3003
3009	Geoff Cameron	Berlin	100	5003	-
3003	Jozy Altridor	Moscow	200	5007	-

22. Multiply the commission of salesman to 7890.80. Round the results.

23. Show the customers names and the referred columns. Using a function, replace the null values to “no referral”. Hint: To able to this you have to convert the referral to a character first by using another function.

Sample Results:

Fabian Johnson	No referral
Nick Rimando	3006
Brad Davis	No referral
Graham Zusi	No referral
Julian Green	3003
Geoff Cameron	No referral
Jozy Altridor	No referral

24. Extract the first 3 characters of the customer name in the customer table.