**SQL Assignment**

**Name: Rahul Babasaheb Desai**

**Batch: September Weekend Batch 2024: DSAIML**

**=====================================================================================**

# Q1. How many rows are there in the table?  
select count(\*) from Customers;  
# Answer: In Customers table we have 5 rows  
#-------------------------------------------  
select count(\*) from Orders;  
# Answer: In Orders table we have 100 rows  
#===========================================  
  
# Q2. Do we have missing values in any column?  
select count(\*) as null\_values  
from Orders  
where OrderId is null  
or CustomerId is null  
or OrderDate is null  
or DeliveryDate is null  
or OrderValuePerQty is null  
or Qty is null;  
#Answer: in Orders table we don't have any missing (or null) value  
#-----------------------------------------  
select count(\*) as null\_values  
from Customers  
where CustomerID is null  
or OnBoardingDate is null  
or RevenueTillDate is null  
or CustomerStatus is null;  
#Answer: in Customers table we don't have any missing (or null) value  
#===========================================  
  
# Q3. Pull only order date and delivery date from the table.  
select OrderDate, DeliveryDate from Orders;  
#===========================================  
  
# Q4. how many distinct customers we have  
select count(distinct CustomerID) as distinct\_customers from Orders;  
#Answer: We have 5 distict customers  
#===========================================

# Q5. What is the minimum and maximum delivery date  
select min(DeliveryDate) as minimum\_delivery\_date,  
max(DeliveryDate) as maximum\_delivery\_date  
from Orders;  
# minimum delivery date: 2024-02-10 and maximum delivery date: 2024-11-28  
#===========================================  
  
# Q6. What is the avg qty per order  
select avg(qty) as average\_qty\_per\_order from orders;  
# 3.4600  
#===========================================  
  
# Q7. Create a new column order value new which is calculated by gving 20% discount  
# on items which are of less than 80 in value and 25% discount on items which are more than $80.  
select \*,  
case  
when OrderValuePerQty<80 then OrderValuePerQty-OrderValuePerQty\*0.2  
when OrderValuePerQty>=80 then OrderValuePerQty-OrderValuePerQty\*0.25  
end as OrderValueNew  
from Orders;  
#===========================================  
  
# Q8. Bring customer status information to order table  
select a.\*,b.CustomerStatus  
from Orders as a  
left join  
Customers as b  
on a.CustomerID=b.CustomerID;  
#===========================================  
  
# Q9. How many orders are there for each customer status?  
select count(\*) as no\_of\_orders,b.CustomerStatus  
from Orders as a  
left join  
Customers as b  
on a.CustomerID=b.CustomerID  
group by 2;  
#===========================================  
  
# Q10. Calculate order gap for each order. (Order gap is the difference in days between two consectutive orders of a customer)  
SELECT  
    OrderID,  
    CustomerID,  
    OrderDate,  
    DeliveryDate,  
    OrderValuePerQty,  
    Qty,  
    DATEDIFF(OrderDate, LAG(OrderDate) OVER (PARTITION BY CustomerID ORDER BY OrderDate)) AS OrderGapDays  
FROM  
    Orders;  
  
#===========================================  
  
# Q11. Rank orders of each customer by their total order value (totalorder value = price per qty \* qty)  
select \*,  
case when OrderValuePerQty then OrderValuePerQty\*Qty end as TotalOrderValue,  
rank() over(partition by CustomerId order by OrderValuePerQty\*Qty desc) as rank1,  
dense\_rank() over(partition by CustomerId order by OrderValuePerQty\*Qty desc) as dense\_rank2  
from Orders;  
#===========================================