Final ASSIGNMENT 5

1. select month(dateentered) month_, count(case when year(dateentered) = 2020 then customerid else null end) as year_2020, count(case when year(dateentered) = 2021 then customerid else null end) as year_2021 from Customers GROUP by month(dateentered) order by month(dateentered);

2.

```
Select * from
(select cat.CategoryID, pro.ProductID,  Sum(ordt.Quantity) As Qnty,
Dense_Rank () Over(Partition by CategoryID Order by Sum(ordt.Quantity) Desc) As D_Rank
from Category As cat
inner join Products As pro
on cat.CategoryID = pro.Category_ID
inner join OrderDetails As ordt
on pro.ProductID = ordt.ProductID
where cat.Active = 1
group by pro.ProductID, cat.CategoryID) new
where D_Rank between 1 and 3
```

3.

```
Select * from
(select cat.CategoryID, pro.ProductID,  Sum(ordt.Quantity) As Qnty,
Dense_Rank () Over(Partition by CategoryID Order by Sum(ordt.Quantity)) As D_Rank
from Category As cat
inner join Products As pro
on cat.CategoryID = pro.Category_ID
inner join OrderDetails As ordt
on pro.ProductID = ordt.ProductID
where cat.Active = 1
group by pro.ProductID, cat.CategoryID) new
where D_Rank = 1
```

4. create view prd_ord5

```
As
(select ordtl.OrderDetailID, ordtl.OrderID, ordtl.ProductID , prdt1.Product, prdt1.Category_ID,
ords.CustomerID from OrderDetails As ordtl
inner join Products As prdt1 on ordtl.ProductID = prdt1.ProductID
inner join Orders As ords on Ords.OrderID = ordtl.OrderID);

create view prd_ord6
As
(select ordtl2.OrderDetailID, ordtl2.OrderID, ordtl2.ProductID , prdt2.Product, prdt2.Category_ID
, ords2.CustomerID from OrderDetails As ordtl2
```

```
inner join Products As prdt2 on ordtl2.ProductID = prdt2.ProductID
inner join Orders As ords2 on Ords2.OrderID = ordtl2.OrderID);


Select top 1 prd_ord5.ProductID , prd_ord6.CustomerID,prd_ord5.Product,prd_ord6.ProductID ,
prd_ord6.Product,count(prd_ord5.ProductID) As number_of_times
from prd_ord5
inner join
prd_ord6
on prd_ord5.OrderID = prd_ord6.OrderID and prd_ord5.ProductID > prd_ord6.ProductID and
prd_ord5.CustomerID = prd_ord6.CustomerID
group by prd_ord5.ProductID ,  prd_ord6.ProductID , prd_ord5.Product, prd_ord6.Product,
prd_ord6.CustomerID
Order by number_of_times Desc
```
Product Purchased together

5. a . Select SUM(OrderDetails.Quantity* Products.Sale_Price) As rev from Orders
inner Join Customers on Customers.CustomerID = Orders.CustomerID
inner Join OrderDetails on OrderDetails.OrderID = Orders.OrderID
inner join Products on Products.ProductID = OrderDetails.ProductID
inner Join Category on Category.CategoryID = Products.Category_ID
where (year(orderDate) = 2021) and (month(orderDate) between 7 and 10)
and (Category.CategoryName = 'Beauty & Hygiene' or Category.CategoryName = 'Beverages');

5. B . Select Sum(products.Sale_Price * orderdetails.Quantity) as
total_sale_from_Beauty_Hygiene_Bevarages_from_new_cust
from Orders
inner Join Customers on Customers.CustomerID = Orders.CustomerID
inner Join OrderDetails on OrderDetails.OrderID = Orders.OrderID
inner join Products on Products.ProductID = OrderDetails.ProductID
inner Join Category on Category.CategoryID = Products.Category_ID
where year(orderDate) = 2021 and month(orderDate) between 7 and 10
and (Category.CategoryName = 'Beauty & Hygiene' or Category.CategoryName = 'Beverages')
and (year(customers.DateEntered) = 2021 and month(customers.DateEntered) between 7 and
10);

6.

Select * , Rank() over(Partition by CategoryName order by Revenue DESC) As Rank_ from (
Select DATEPART(q, OrderDate) As Qtr, CategoryName, Sum(products.Sale_Price * Quantity)
As revenue from Orders
```
```

```
inner Join Customers on Customers.CustomerID = Orders.CustomerID
inner Join OrderDetails on OrderDetails.OrderID = Orders.OrderID
inner join Products on Products.ProductID = OrderDetails.ProductID
inner Join Category on Category.CategoryID = Products.Category_ID
where year(Orders.OrderDate) = 2020
Group by DATEPART(q, OrderDate), CategoryName)c
Order By Qtr;
```

7.

A.
```
SELECT * FROM
(SELECT
P.Category_ID,O.ShipperID,AVG(DATEDIFF(DAY,O.OrderDate,O.DeliveryDate)
) AS 'Delivery Period',
ROW_NUMBER() OVER(PARTITION BY P.Category_ID ORDER BY
AVG(DATEDIFF(DAY,O.OrderDate,O.DeliveryDate))) AS 'Rank'
FROM Products AS P
JOIN OrderDetails AS OD ON P.ProductID=OD.ProductID
JOIN Orders AS O ON OD.OrderID=O.OrderID
WHERE YEAR(O.OrderDate)=2021
GROUP BY P.Category_ID,O.ShipperID
)DT
WHERE DT.Rank<=3;
```

b) select top 3 Companyname orders o inner join orderdetails OD on o.orderID = OD.orderID left join shippers S on o.shipperId = s.shipperID

where year(OrderDate) in (Select Year (MAx(OrderDate))from orders)

group by O.shipperID, companyname

Order by sum(quantity) desc

8. with cte1 as (select customerid, count(orderid) as total_orders, sum(total_order_amount) as total_purchase_Amt

from orders
where Date_PART('YEAR', cast(orderdate as date)) = '2021'
GROUP by customerid),

cte2 as (select *, DENSE_RANK() over (order by total_orders desc) as order_rank,
DENSE_RANK() over (order by total_purchase_Amt desc) as purchase_rank
from cte1)

a) select B.*, A.total_orders, A.order_rank from cte2 as a left join customers as b on a.customerid = b.customerid
   where order_rank <=25
   order by order_rank;

b) select B.*, A.total_purchase_Amt, A.purchase_rank from cte2 as a left join customers as b on a.customerid = b.customerid
   where purchase_rank <=25
   order by purchase_rank;

9.
Create View I_View
As
(Select * from
(select *, Dense_rank() OVER (partition by CustomerId Order by OrderDate Desc) As rank_
from Orders) c
where rank_ = 1 or rank_ = 2)

Select CustomerID, Case
When Datediff(day, Min(OrderDate) , Max(OrderDate)) < 5 then 'Frequent'
Else 'Infrequent'
End
As Tag_
from I_View
group by CustomerID

10. a
Select *, Avg(monthly_total) Over (Partition by Category_ID order by months)As Cum_avg from
(Select C.Category_ID, Month(OrderDate) As months, SUM(Sale_Price* Quantity) As
monthly_total from Orders As A
inner Join OrderDetails As B
on A.OrderID = B.OrderID

```
inner join Products As C on C.ProductID = B.ProductID
where Year(OrderDate) = 2021
Group by C.Category_ID, Month(OrderDate))c

10.B
Select *, Avg(monthly_total) Over (Partition by CustomerID order by months)As Cum_avg from
(Select A.CustomerID, Month(OrderDate) As months, SUM(Total_order_amount) As
monthly_total from Orders As A
inner Join OrderDetails As B
on A.OrderID = B.OrderID
inner join Products As C on C.ProductID = B.ProductID
where Year(OrderDate) = 2021
Group by A.CustomerID, Month(OrderDate))c

11.
Select * , AVG(Total_Order_amount)
Over ( Partition by CustomerId Order by orderdate Rows 2 PRECEDING  ) As three_day_rolling
from Orders


12.
Create View
PVT_view
As
(Select A.CustomerID, A.Total_order_amount,B.PaymentType  from Orders As A inner join
Payments AS B on A.PaymentID = B.PaymentID)

Select CustomerID,[Debit Card],[POD],[PayPal], [Credit Card],[Wallet],[Net banking] from
PVT_view
Pivot
(Sum(Total_order_amount) for PaymentType in ([Debit Card],[POD],[PayPal],[Credit
Card],[Wallet],[Net banking])) As Pvt


13.
Create Procedure Filter_ @X float, @y int
As
begin
Select * from Orders
where OrderID < @X and year(OrderDate) = @Y
End

Exec Filter_ 77000000, 2021
```

Q6.1. What is the possible data type of the column ‘COL1’?

NVarchar

2. What will the output of the following SQL statements

a. ‘SELECT COUNT (*) AS ENTRIES FROM TABLE;’

⇒ 7

b. ‘SELECT COUNT(COL1) AS ENTRIES FROM TABLE;’

⇒ 6

c. ‘SELECT COUNT (DISTINCT COL1) AS ENTRIES FROM TABLE;’

⇒ 6