Write SQL Queries for following Schema

Table Schema

Customers (

```
CustomerID          INT PRIMARY KEY,
Name                VARCHAR(50) NOT NULL,
JoinDate            DATE NOT NULL);
```

Products (

```
ProductID           INT PRIMARY KEY,
ProductName         VARCHAR(100) NOT NULL,
Category            VARCHAR(50) NOT NULL,
Price               DECIMAL(10,2) NOT NULL,
StockQuantity       INT NOT NULL);
```

Orders (

```
OrderID             INT PRIMARY KEY,
CustomerID          INT FOREIGN KEY REFERENCES Customers(CustomerID),
ProductID           INT FOREIGN KEY REFERENCES Products(ProductID),
OrderDate           DATE NOT NULL,
Quantity            INT NOT NULL,
Price                DECIMAL(10,2) NOT NULL
Amount               DECIMAL(10,2) NOT NULL,
ShippedDate         DATE NULL
OrderStatus         VARCHAR(20) CHECK (OrderStatus IN ('Pending', 'Shipped', 'Delivered', 'Cancelled')));
```

Payments (

```
PaymentID           INT PRIMARY KEY
OrderID             INT FOREIGN KEY REFERENCES Orders (OrderID)
PaymentDate         DATE NOT NULL
PaymentAmount       Decimal(10,2) NOT NULL);
```

Write મસ્ત Queries for the following

1. **List Customers with total outstanding amount**

Customer Name	Outstanding Amount

```
SELECT C.CustomerID, C.Name, SUM(O.Amount - COALESCE(P.PaidAmount, 0)) AS OutstandingAmount
FROM Customers C
JOIN Orders O ON C.CustomerID = O.CustomerID
LEFT JOIN (
  SELECT OrderID, SUM(PaymentAmount) AS PaidAmount
  FROM Payments
  GROUP BY OrderID
) P ON O.OrderID = P.OrderID
WHERE (O.Amount - COALESCE(P.PaidAmount, 0)) > 0
GROUP BY C.CustomerID, C.Name;
```

OR

```
select c.Name, sum(amount) from [dbo].[Customers] c left join
[dbo].[Orders] o on c.CustomerID=o.CustomerID
```

```
left join [dbo].[Payments] p on o.orderid=p.orderid where
p.paymentamount is null
group by c.Name having sum(amount)>0
```

2. List Customers who placed more than 2 orders in a single day of a same product

Customer Name	Product Name	Total Order	Total Quantity

```
SELECT C.Name AS CustomerName, P.ProductName, COUNT(O.OrderID) AS TotalOrders,
 SUM(O.Amount) AS TotalAmount, SUM(O.Quantity) AS TotalQuantity
FROM Orders O
JOIN Customers C ON O.CustomerID = C.CustomerID
JOIN Products P ON O.ProductID = P.ProductID
GROUP BY C.Name, P.ProductName, O.OrderDate
HAVING COUNT(O.OrderID) > 2;
```

3. List Orders which are delivered but payment not received

Customer Name	Product Name	Order ID	Order Date	Amount

```
SELECT C.Name AS CustomerName, P.ProductName, O.OrderID, O.OrderDate, O.Amount
FROM Orders O
JOIN Customers C ON O.CustomerID = C.CustomerID
JOIN Products P ON O.ProductID = P.ProductID
LEFT JOIN Payments Pay ON O.OrderID = Pay.OrderID
WHERE O.OrderStatus = 'Delivered' AND Pay.OrderID IS NULL;
```

4. List All Products whose price is more than Product “Keyboard”

Product Name	Price

```
SELECT ProductName, Price
FROM Products
WHERE Price > (SELECT Price FROM Products WHERE ProductName = 'Keyboard');
```

5. List Customers which have not placed a single order

Customer ID	Customer Name	Join Date

```
SELECT C.CustomerID, C.Name, C.JoinDate
FROM Customers C
LEFT JOIN Orders O ON C.CustomerID = O.CustomerID
WHERE O.OrderID IS NULL;
```

6. List Products where StockQuantity is less than Pending Order Total Quantity

Category	Product Name	Pending Orders	Pending to Ship Quantity	Stock Quantity

```
SELECT
   P.Category,
   P.ProductName,
   COUNT(O.OrderID) AS PendingOrders,
   SUM(O.Quantity) AS PendingToShipQuantity,
   P.StockQuantity
FROM
   Products P
IINER JOIN
   Orders O ON P.ProductID = O.ProductID
WHERE
```

```
    O.OrderStatus = 'Pending'
GROUP BY
    P.ProductID, P.Category, P.ProductName, P.StockQuantity
HAVING
    SUM(O.Quantity) > P.StockQuantity;
```

7. Category Wise Sales Summary (Include all the category even if with zero orders)

Category	No. of Customers	No. of Orders	No. of Products	Total Quantity	Total Order Amount	Total Received Payment

```
SELECT
    P.Category,
    COUNT(DISTINCT P.ProductID) AS NoOfProducts,
    COUNT(DISTINCT O.OrderID) AS NoOfOrders,
    COUNT(DISTINCT O.CustomerID) AS NoOfCustomers,
    COALESCE(SUM(O.Quantity), 0) AS TotalQuantity,
    COALESCE(SUM(O.Amount), 0) AS TotalOrderAmount,
    COALESCE(SUM(PAY.PaymentAmount), 0) AS TotalReceivedPayment
FROM
    Products P
LEFT JOIN
    Orders O ON P.ProductID = O.ProductID
LEFT JOIN
    Payments PAY ON O.OrderID = PAY.OrderID
GROUP BY
    P.Category
ORDER BY
    P.Category;
```

8. List Customer with Order detail who has placed order of same product again within 7 days.

Sr.	Customer Name	Product	OrderID	Quantity

```
SELECT
    ROW_NUMBER() OVER (ORDER BY O2.OrderDate) AS SrNo,
    C.Name AS CustomerName,
    P.ProductName,
    O2.OrderID,
    O2.Quantity
FROM
    Orders O1
JOIN
    Orders O2
    ON O1.CustomerID = O2.CustomerID
    AND O1.ProductID = O2.ProductID
    AND O1.OrderID < O2.OrderID
    AND DATEDIFF(DAY, O1.OrderDate, O2.OrderDate) BETWEEN 1 AND 7
JOIN
    Customers C ON O2.CustomerID = C.CustomerID
JOIN
    Products P ON O2.ProductID = P.ProductID;
```

9. Product Wise Total Orders, Lowest Price, Highest Price & Average Price

Sr.	Category	Product	Total Orders	Lowest Price	Highest Price	Average Price

```
SELECT
   ROW_NUMBER() OVER (ORDER BY P.Category, P.ProductName) AS SrNo,
   P.Category,
   P.ProductName,
   COUNT(O.OrderID) AS TotalOrders,
   MIN(O.Price) AS LowestPrice,
   MAX(O.Price) AS HighestPrice,
   ROUND(AVG(O.Price), 2) AS AveragePrice
FROM
   Products P
LEFT JOIN
   Orders O ON P.ProductID = O.ProductID
GROUP BY
   P.Category, P.ProductName
ORDER BY
   P.Category, P.ProductName;
```

10. List Date Wise Pending Order and Pending Ordered Quantity Product Wise between fromDate and toDate

Sr.	Date	Product	Total Pending Orders	Total Pending Quantity

```
SELECT
   ROW_NUMBER() OVER (ORDER BY O.OrderDate, P.ProductName) AS SrNo,
   O.OrderDate AS Date,
   P.ProductName AS Product,
   COUNT(O.OrderID) AS TotalPendingOrders,
   SUM(O.Quantity) AS TotalPendingQuantity
FROM
   Orders O
JOIN
   Products P ON O.ProductID = P.ProductID
WHERE
   O.OrderStatus = 'Pending'
   AND O.OrderDate BETWEEN '2024-01-01' AND '2024-12-31'
GROUP BY
   O.OrderDate, P.ProductName
ORDER BY
   O.OrderDate, P.ProductName;
```