```
/* Lab 5-1 Solution*/

USE Kashyap
CREATE FUNCTION getTotalSales (@year int,@month int)
RETURNS INT
AS
BEGIN
      DECLARE @TotSale INT;

      SELECT @TotSale = SUM(TotalDue)
      FROM AdventureWorks2008R2.Sales.SalesOrderHeader
      WHERE YEAR (OrderDate) = @year AND MONTH (OrderDate) = @month
      GROUP BY YEAR (OrderDate),  MONTH (OrderDate)
      RETURN ISNULL(@TotSale,0)
END

USE   Kashyap
SELECT dbo.[getTotalSales](2007,8) as TotalSales
DROP FUNCTION getTotalSales

-----------------------------------------------------------------
/* Lab 5-2 Solution*/

USE Kashyap

CREATE TABLE DateRange
(DateID INT IDENTITY,
DateValue DATE,
DayOfWeek INT,
Month INT,
);

CREATE PROCEDURE PopulateDateRange
      @startDate DATE,
      @numofdates INT
AS
BEGIN
      DECLARE @count INT = 0;
      WHILE @count<@numofdates
      BEGIN
            DECLARE @dateSet DATE=dateadd(day,@count,@startDate);
            DECLARE @dayofweek INT=datepart(dw,@dateSet) ;
            DECLARE @month INT=month(@dateSet);

            INSERT INTO DateRange
                  VALUES(@dateSet,
                              @dayofweek,
                              @month
                              );
            SET @count+=1;
      END
END

---Execute the Procedure
EXEC PopulateDateRange '2022-04-06',10
--- Check the table
SELECT * FROM DateRange
```

```
DROP TABLE DateRange
DROP PROCEDURE PopulateDateRange


-----------------------------------------------------------------
/* Lab 5-3 Solution*/

CREATE TABLE Customer
(CustomerID VARCHAR(20) PRIMARY KEY,
 CustomerLName VARCHAR(30),
 CustomerFName VARCHAR(30),
 CustomerStatus VARCHAR(10));

CREATE TABLE SaleOrder
(OrderID INT IDENTITY PRIMARY KEY,
 CustomerID VARCHAR(20) REFERENCES Customer(CustomerID),
 OrderDate DATE,
 OrderAmountBeforeTax INT);

CREATE TABLE SaleOrderDetail
(OrderID INT REFERENCES SaleOrder(OrderID),
 ProductID INT,
 Quantity INT,
 UnitPrice INT,
 PRIMARY KEY (OrderID, ProductID));

 SELECT * FROM Customer

 CREATE TRIGGER tr_CustomerStatus
 ON dbo.SaleOrder
 AFTER INSERT,UPDATE,DELETE AS
 BEGIN
      DECLARE @total money = 0;
      DECLARE @custid varchar(20);
      DECLARE @status varchar(10);

      SELECT @custid = isnull (i.CustomerID, d.CustomerID)
            FROM inserted i full join deleted d
            ON i.CustomerID = d.CustomerID;

      SELECT @total = sum(OrderAmountBeforeTax)
            FROM saleOrder
            WHERE CustomerID = @custid;

      IF @total > 5000
            SET @status = 'Preferred'
      ELSE
            SET @status = 'Normal';

      UPDATE Customer
            SET CustomerStatus = @status
            WHERE CustomerID = @custid
END


INSERT Customer VALUES ('100','Scott','Dora','Normal');
INSERT SaleOrder VALUES ('100','2022-04-04',1000);
SELECT * FROM Customer;
SELECT * FROM SaleOrder
```

```
INSERT SaleOrder VALUES ('100','2022-04-05',5000);
SELECT * FROM Customer;
SELECT * FROM SaleOrder

UPDATE SaleOrder SET OrderAmountBeforeTax = 1000
WHERE CustomerID = '100' and OrderDate = '2022-04-05';
SELECT * FROM Customer;
SELECT * FROM SaleOrder;

INSERT SaleOrder VALUES ('100','2022-04-04',10000);
SELECT * FROM Customer;
SELECT * FROM SaleOrder;

DELETE SaleOrder WHERE CustomerID = '100' and OrderDate = '2022-04-04';
SELECT * FROM Customer;
SELECT * FROM SaleOrder;

DROP TRIGGER tr_CustomerStatus
DROP TABLE SaleOrderDetail
DROP TABLE SaleOrder
DROP TABLE Customer
```