--Part.D.D1

CREATE PROCEDURE sp\_emp\_info

(@paraempid char(2) = NULL)

AS

SELECT EmployeeID, LastName, FirstName, Phone

FROM Employees

WHERE EmployeeID = @paraempid

GO

--Use below query to test the stored procedure

EXEC sp\_emp\_info 7

--Part.D.D2

CREATE PROCEDURE sp\_orders\_by\_dates

(@paradate1 datetime,

@paradate2 datetime)

AS

SELECT Orders.OrderID,

Orders.CustomerID,

Customers.CompanyName,

Shippers.CompanyName AS ShipperName,

Orders.ShippedDate

FROM Orders

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

JOIN Shippers ON Shippers.ShipperID = Orders.ShipperID

WHERE ShippedDate BETWEEN @paradate1 and @paradate2

ORDER BY Orders.OrderID

GO

--Use below query to test the stored procedure

EXEC sp\_orders\_by\_dates '1991-01-01' , '1991-12-31'

--Part.D.D3

CREATE PROCEDURE sp\_products

(@productname varchar(40),

@month varchar(10),

@year int)

AS

SELECT Products.ProductName,

Products.UnitPrice,

Products.UnitsInStock,

Suppliers.Name

FROM Products

JOIN Suppliers ON Products.SupplierID = Suppliers.SupplierID

JOIN OrderDetails ON Products.ProductID = OrderDetails.ProductID

JOIN Orders ON Orders.OrderID = OrderDetails.OrderID

WHERE Products.ProductName LIKE '%'+@productname+'%'

AND Datename(month, Orders.OrderDate) = @month

AND Datename(Year, Orders.OrderDate) = @year

GO

--Use below query to test the stored procedure

EXEC sp\_products '%tofu%', 'December', 1992

--Part.D.D4

CREATE PROCEDURE sp\_unit\_prices

(@unitprice1 money,

@unitprice2 money)

AS

SELECT ProductID,

ProductName,

EnglishName,

UnitPrice

FROM Products

WHERE UnitPrice BETWEEN @unitprice1 AND @unitprice2

ORDER BY ProductID

GO

--Use below query to test the stored procedure

EXEC sp\_unit\_prices 5.50, 8.00

--Part.D.D5

CREATE PROCEDURE sp\_customer\_city

(@city varchar(15))

AS

SELECT CustomerID,

CompanyName,

Address,

City,

Phone

FROM Customers

WHERE City = @city

ORDER BY CustomerID

GO

--Use below query to test the stored procedure

EXEC sp\_customer\_city 'Paris'

--Part.D.D6

CREATE PROCEDURE sp\_reorder\_qty

(@qty smallint)

AS

SELECT Products.ProductID,

Products.ProductName,

Suppliers.Name,

Products.UnitsInStock,

Products.ReorderLevel

FROM Products

JOIN Suppliers ON Products.SupplierID = Suppliers.SupplierID

WHERE (Products.UnitsInStock - Products.ReorderLevel) < @qty

ORDER BY Products.ProductID

GO

--Use below query to test the stored procedure

EXEC sp\_reorder\_qty 9

--Part.D.D7

CREATE PROCEDURE sp\_shipping\_date

(@shippeddate smalldatetime)

AS

SELECT Orders.OrderID,

Customers.CompanyName AS CustomerName,

Shippers.CompanyName AS ShipperName,

Orders.OrderDate,

Orders.ShippedDate

FROM Shippers

JOIN Orders ON Shippers.ShipperID = Orders.ShipperID

JOIN Customers ON Orders.CustomerID = Customers.CustomerID

WHERE DATEADD(DAY,10, Orders.OrderDate) = @shippeddate

ORDER BY Orders.OrderID

GO

--Use below query to test the stored procedure

EXEC sp\_shipping\_date '1993-11-29'

--Part.D.D8

CREATE PROCEDURE sp\_del\_inactive\_cust

AS

DELETE

FROM Customers

WHERE CustomerID NOT IN

(SELECT Orders.CustomerID

FROM Orders)

GO

--Use below query to test the stored procedure

EXEC sp\_del\_inactive\_cust

--Part.D.D9

CREATE TRIGGER tr\_check\_qty

ON OrderDetails

FOR UPDATE

AS

IF (SELECT Quantity FROM inserted OrderDetails) > (SELECT Products.UnitsInStock FROM Products

JOIN inserted OrderDetails ON OrderDetails.ProductID = Products.ProductID)

BEGIN

PRINT 'Quantity ordered exceeds units in stock'

ROLLBACK TRANSACTION

END

--Use below query to test the stored procedure

UPDATE OrderDetails

SET Quantity = 40

WHERE OrderID = 10044

AND ProductID = 77

--Part.D.D10

CREATE TRIGGER tr\_insert\_shippers

ON Shippers

INSTEAD OF INSERT

AS

IF EXISTS (SELECT CompanyName FROM inserted)

BEGIN

PRINT 'Company name already exists'

END

ELSE

BEGIN

INSERT INTO Shippers

SELECT \*

FROM inserted

END

--Use below query to test the stored procedure

INSERT Shippers

VALUES (4,'Federal Shipping')