-- D.1:

```
CREATE PROCEDURE     sp_emp_info
(@emp_id_param       int)
AS
     SELECT          EmployeeID,
                     LastName,
                     FirstName,
                     Phone
     FROM            Employees
     WHERE           EmployeeID = @emp_id_param
GO
```

-- D.2:

```
CREATE PROCEDURE          sp_orders_dates
(    @start_date_param         smalldatetime,
     @end_date_param           smalldatetime        )
AS
     SELECT o.OrderID,
                o.CustomerID,
                c.CompanyName,
                s.CompanyName AS    ShipperName,
                o.ShippedDate
     FROM       Orders o
     JOIN       Customers   c ON  o.CustomerID =     c.CustomerID
     JOIN       Shippers    s ON  o.ShipperID  =     s.ShipperID
     WHERE      o.ShippedDate BETWEEN @start_date_param AND @end_date_param
GO
```

-- D.3:

```sql
CREATE PROCEDURE                sp_products
(       @product_name_param         nvarchar(40),
        @month_param                varchar(12),
        @year_param                 int      )
AS
        SELECT  p.ProductName,
                p.UnitPrice,
                p.UnitsInStock,
                s.Name
        FROM    Products        p
        JOIN    Suppliers       s       ON      p.SupplierID    =       s.SupplierID
        JOIN    OrderDetails    od      ON      p.ProductID     =       od.ProductID
        JOIN    Orders          o       ON      od.OrderID      =       o.OrderID
        WHERE   p.ProductName LIKE @product_name_param
        AND
        DATENAME(MONTH,o.OrderDate) = @month_param
        AND
        DATEPART(YEAR, o.OrderDate) = @year_param
GO
```

-- D.4:

```sql
CREATE PROCEDURE        sp_unit_prices
(       @unit_price1_param  money,
        @unit_price2_param  money           )
AS
        SELECT          ProductID,
                        ProductName,
                        EnglishName,
                        UnitPrice
        FROM            Products
        WHERE           UnitPrice Between @unit_price1_param AND @unit_price2_param
GO
```

-- D.5:

```sql
CREATE PROCEDURE     sp_customers_city
(    @city_param     nvarchar(15)     )
AS
     SELECT     CustomerID,
                CompanyName,
                Address,
                City,
                Phone
     FROM       Customers
     WHERE      City =     @city_param
GO
```

-- D.6:

```sql
CREATE PROCEDURE sp_reorder_qty
(    @unit_value_param          smallint   )
AS
     SELECT     p.ProductID,
                p.ProductName,
                s.Name,
                p.UnitsInStock,
                p.ReorderLevel
     FROM       Products   p
     JOIN       Suppliers s     ON    p.SupplierID = s.SupplierID
     WHERE      (p.UnitsInStock - p.ReorderLevel) < @unit_value_param
GO
```

-- D.7:

```sql
CREATE PROCEDURE sp_shipping_date
(    @ship_date_param      smalldatetime   )
AS
    SELECT     o.OrderID,
               c.CompanyName          AS   CustomerName,
               s.CompanyName          AS   ShipperName,
               o.OrderDate,
               o.ShippedDate
    FROM       Orders     o
    JOIN       Customers c    ON    o.CustomerID = c.CustomerID
    JOIN       Shippers  s    ON    o.ShipperID  = s.ShipperID
    WHERE      DATEADD(DAY,10,o.OrderDate) = @ship_date_param
GO
```

-- D.8:

```sql
CREATE PROCEDURE        sp_del_inactive_cust
AS
    DELETE FROM     Customers
    WHERE           CustomerID NOT IN
                              (SELECT    CustomerID
                               FROM      Orders)
GO
```

-- D.9:

```sql
CREATE TRIGGER      tr_check_qty
ON                  OrderDetails
FOR                 UPDATE
AS
IF    (SELECT       Quantity
       FROM         inserted) >
                             (SELECT       p.UnitsInStock
                              FROM         Products p
                              JOIN         inserted i   ON    p.ProductID = i.ProductID)
         BEGIN
                PRINT 'UIS is less than Qty'
                ROLLBACK TRANSACTION
         END
```

-- D.10:

```sql
CREATE TRIGGER           tr_insert_shippers
ON                       Shippers
INSTEAD OF INSERT
AS
IF ((SELECT      CompanyName
       FROM      inserted) = (SELECT      s.CompanyName
                              FROM         Shippers s
                              JOIN         inserted i   ON s.CompanyName = i.CompanyName
                              WHERE        s.CompanyName = i.CompanyName))
           BEGIN
                  PRINT 'This company is repetetive!'
           END
       ELSE
           BEGIN
                  INSERT INTO   Shippers
                  SELECT        ShipperID, CompanyName
                  FROM          inserted
           END
```