```sql
--This is my personal work uploaded on github only for "Celebal Technology" access using inappropriately can cause Copyright infringement,you need my permission before fair use--


-- Drop existing procedures
IF OBJECT_ID('InsertOrderDetails', 'P') IS NOT NULL DROP PROCEDURE InsertOrderDetails;
GO

IF OBJECT_ID('UpdateOrderDetails', 'P') IS NOT NULL DROP PROCEDURE UpdateOrderDetails;
GO

IF OBJECT_ID('GetOrderDetails', 'P') IS NOT NULL DROP PROCEDURE GetOrderDetails;
GO

IF OBJECT_ID('DeleteOrderDetails', 'P') IS NOT NULL DROP PROCEDURE DeleteOrderDetails;
GO

-- Drop existing functions
IF OBJECT_ID('FormatDateMMDDYYYY', 'FN') IS NOT NULL DROP FUNCTION FormatDateMMDDYYYY;
GO

IF OBJECT_ID('FormatDateYYYYMMDD', 'FN') IS NOT NULL DROP FUNCTION FormatDateYYYYMMDD;
GO

-- Drop existing views
IF OBJECT_ID('vwCustomerOrders', 'V') IS NOT NULL DROP VIEW vwCustomerOrders;
GO

IF OBJECT_ID('vwCustomerOrdersYesterday', 'V') IS NOT NULL DROP VIEW vwCustomerOrdersYesterday;
GO

IF OBJECT_ID('MyProducts', 'V') IS NOT NULL DROP VIEW MyProducts;
GO

-- Drop existing triggers
IF OBJECT_ID('trgDeleteOrderDetails', 'TR') IS NOT NULL DROP TRIGGER trgDeleteOrderDetails;
GO

IF OBJECT_ID('trgCheckInventory', 'TR') IS NOT NULL DROP TRIGGER trgCheckInventory;
GO

-- Create InsertOrderDetails procedure
```

```sql
CREATE PROCEDURE InsertOrderDetails
    @OrderID INT,
    @ProductID INT,
    @UnitPrice MONEY = NULL,
    @Quantity INT,
    @Discount DECIMAL(5, 2) = 0
AS
BEGIN
    IF @UnitPrice IS NULL
    BEGIN
        SELECT @UnitPrice = ListPrice
        FROM Production.Product
        WHERE ProductID = @ProductID;
    END

    BEGIN TRY
        BEGIN TRANSACTION;

        INSERT INTO Sales.SalesOrderDetail (SalesOrderID, ProductID,
          UnitPrice, OrderQty, UnitPriceDiscount)
        VALUES (@OrderID, @ProductID, @UnitPrice, @Quantity,
          @Discount);

        UPDATE Production.ProductInventory
        SET Quantity = Quantity - @Quantity
        WHERE ProductID = @ProductID;

        IF EXISTS (SELECT 1 FROM Production.ProductInventory WHERE
          ProductID = @ProductID AND Quantity < 0)
        BEGIN
            RAISERROR('Not enough stock', 16, 1);
            ROLLBACK TRANSACTION;
            RETURN;
        END

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        ROLLBACK TRANSACTION;
        RAISERROR('Failed to place the order. Please try again.', 16,
          1);
    END CATCH
END;
GO

-- Create UpdateOrderDetails procedure
CREATE PROCEDURE UpdateOrderDetails
    @OrderID INT,
    @ProductID INT,
    @UnitPrice MONEY = NULL,
    @Quantity INT = NULL,
    @Discount DECIMAL(5, 2) = NULL
AS
```

```sql
BEGIN
    -- Variable declarations to store original values
    DECLARE @OriginalUnitPrice MONEY;
    DECLARE @OriginalQuantity INT;
    DECLARE @OriginalDiscount DECIMAL(5, 2);

    -- Fetch original values if input values are NULL
    SELECT
        @OriginalUnitPrice = UnitPrice,
        @OriginalQuantity = OrderQty,
        @OriginalDiscount = UnitPriceDiscount
    FROM
        Sales.SalesOrderDetail
    WHERE
        SalesOrderID = @OrderID
        AND ProductID = @ProductID;

    -- If input parameters are NULL, use original values
    SET @UnitPrice = ISNULL(@UnitPrice, @OriginalUnitPrice);
    SET @Quantity = ISNULL(@Quantity, @OriginalQuantity);
    SET @Discount = ISNULL(@Discount, @OriginalDiscount);

    -- Update the order details
    UPDATE Sales.SalesOrderDetail
    SET
        UnitPrice = @UnitPrice,
        OrderQty = @Quantity,
        UnitPriceDiscount = @Discount
    WHERE
        SalesOrderID = @OrderID
        AND ProductID = @ProductID;

    -- Adjust inventory
    DECLARE @QuantityDifference INT = @Quantity - @OriginalQuantity;
    UPDATE Production.ProductInventory
    SET Quantity = Quantity - @QuantityDifference
    WHERE ProductID = @ProductID;
END;
GO

-- Create GetOrderDetails procedure
CREATE PROCEDURE GetOrderDetails
    @OrderID INT
AS
BEGIN
    IF NOT EXISTS (SELECT 1 FROM Sales.SalesOrderDetail WHERE
      SalesOrderID = @OrderID)
    BEGIN
        RAISERROR('The OrderID %d does not exist', 16, 1, @OrderID);
        RETURN;
    END

    SELECT *
```

```
143     FROM Sales.SalesOrderDetail
144     WHERE SalesOrderID = @OrderID;
145 END;
146 GO
147
148 -- Create DeleteOrderDetails procedure
149 CREATE PROCEDURE DeleteOrderDetails
150     @OrderID INT,
151     @ProductID INT
152 AS
153 BEGIN
154     IF NOT EXISTS (SELECT 1 FROM Sales.SalesOrderDetail WHERE
          SalesOrderID = @OrderID AND ProductID = @ProductID)
155     BEGIN
156         PRINT 'Invalid parameters';
157         RETURN -1;
158     END
159
160     DELETE FROM Sales.SalesOrderDetail
161     WHERE SalesOrderID = @OrderID AND ProductID = @ProductID;
162 END;
163 GO
164
165 -- Create FormatDateMMDDYYYY function
166 CREATE FUNCTION FormatDateMMDDYYYY (@date DATETIME)
167 RETURNS VARCHAR(10)
168 AS
169 BEGIN
170     RETURN CONVERT(VARCHAR(10), @date, 101);
171 END;
172 GO
173
174 -- Create FormatDateYYYYMMDD function
175 CREATE FUNCTION FormatDateYYYYMMDD (@date DATETIME)
176 RETURNS VARCHAR(8)
177 AS
178 BEGIN
179     RETURN CONVERT(VARCHAR(8), @date, 112);
180 END;
181 GO
182
183 -- Create vwCustomerOrders view
184 CREATE VIEW vwCustomerOrders
185 AS
186 SELECT
187     Name AS CompanyName,
188     SOH.SalesOrderID AS OrderID,
189     SOH.OrderDate,
190     SOD.ProductID,
191     P.Name AS ProductName,
192     SOD.OrderQty AS Quantity,
193     SOD.UnitPrice,
194     SOD.UnitPrice * SOD.OrderQty AS TotalPrice
```

```
195 FROM
196     Sales.Customer AS C
197 JOIN
198     Sales.SalesOrderHeader AS SOH ON C.CustomerID = SOH.CustomerID
199 JOIN
200     Sales.SalesOrderDetail AS SOD ON SOH.SalesOrderID =
          SOD.SalesOrderID
201 JOIN
202     Production.Product AS P ON SOD.ProductID = P.ProductID;
203 GO
204
205 -- Create vwCustomerOrdersYesterday view
206 CREATE VIEW vwCustomerOrdersYesterday
207 AS
208 SELECT *
209 FROM vwCustomerOrders
210 WHERE OrderDate = CONVERT(DATE, GETDATE() - 1);
211 GO
212
213 -- Create MyProducts view
214 CREATE VIEW MyProducts
215 AS
216 SELECT
217     P.ProductID,
218     P.Name AS ProductName,
219     P.ListPrice AS UnitPrice,
220     V.Name AS CompanyName,
221     PC.Name AS CategoryName
222 FROM
223     Production.Product P
224 JOIN
225     Purchasing.ProductVendor PV ON P.ProductID = PV.ProductID
226 JOIN
227     Purchasing.Vendor V ON PV.BusinessEntityID = V.BusinessEntityID
228 JOIN
229     Production.ProductSubcategory PSC ON P.ProductSubcategoryID =
          PSC.ProductSubcategoryID
230 JOIN
231     Production.ProductCategory PC ON PSC.ProductCategoryID =
          PC.ProductCategoryID
232 WHERE
233     P.DiscontinuedDate IS NULL;
234 GO
235
236 -- Create trgDeleteOrderDetails trigger
237 CREATE TRIGGER trgDeleteOrderDetails
238 ON Sales.SalesOrderHeader
239 INSTEAD OF DELETE
240 AS
241 BEGIN
242     DELETE FROM Sales.SalesOrderDetail
243     WHERE SalesOrderID IN (SELECT SalesOrderID FROM deleted);
244
```

```
245     DELETE FROM Sales.SalesOrderHeader
246     WHERE SalesOrderID IN (SELECT SalesOrderID FROM deleted);
247 END;
248 GO
249
250 --This is my personal work uploaded on github only for "Celebal
      Technology" access using inappropriately can cause Copyright
      infringement,you need my per
```