CELEBAL TECHNOLOGIES

Name - Aditya Kumar
Student_Id:CT_CSI_SQ_4665
Domain:SQL
Email ID: adityakrraghopur@gmail.com

LEVEL B TASK

Stored Procedures:

1. InsUnderDetach

- Parameters
  - @OrderID: The ID of the sales order.
  - @ProductID: The ID of the product being ordered.
  - @Priority: The order priority.
  - @Discount: The discount applied to the product.

- Logic:

  I. It declares a variable @InStock to store the available quantity of the product.
  II. It checks if the product has sufficient stock (AvailableQuantity) using a query to Production.ProductionInventory.
  III. If enough stock exists (@InStock >= @Quantity):

  It inserts a new record into Sales.OrderDetail with the provided information and the current unit price obtained using the GetPrice function.

  It updates the AvailableQuantity in Production>ProductInventory by subtracting the ordered quantity.

  IV. If there's insufficient stock, it raises an error message and exits the procedure.

2. UpdateOrderStatus

- Parameters:
  - @OrderID: The ID of the sales order.
  - @ProductID:The ID of the product to update.
  - @NewQuantity:The new quantity for the product.
  - @NewDiscount:The new discount for the product.

- Logic:

1. It declares variables to store the current quantity,unit price and discount.
2. It retrieves the current values for the specified product within the order using a query to Sales.OrderDetail.’
3. It updates the Sales.OrderDetail table using an UPDATE statement:
   - It sets the Quantity to either the provided @NewQunantity or the current value if @NewQuantity is NULL.
   - It retrieves the current unit price using the GetPrice function id @ProductID is provided, otherwise, it keeps the existing value.
   - It sets the Discount to either the provided @NewDiscount or the current value if @NewDiscount is NULL.

3. GetOrderDetails

- This stored procedure retrieves all order details for a specific order.

  - Parameter:
    - @OrderID: The ID of the sales order.
  - Logic:
    1. IT checks if any order details exist for the given order ID using a query to Sales.OrderDetail.
    2. If no records are found, it prints an error message and exits the procedure.
    3. If records exist, it selects all columns from Sales.OrderDetail for the specified order ID.

4. DeleteOrderDetail

- This stored procedure deletes an order detail record for a specific product within an order.

  - Parameters:
    - @OrderID: The ID of the sales order.
    - @ProductID:The ID of the product to delete.
  - Logic:

    1. It checks if the order detail record exists using a query to Sles.OrderDetail.
    2. If the record is not found,it raises an error message and exits the procedure.
    3. If the record exists, it deletes the record from Sales.OrderDetail for the specified order and product.

Functions:

1. FormatDate
    - This function formats a date value into a “MM/DD/YYYY” string.

- Parameters:
    - @date:The date value to be formatted.
- Logic:

    1. It declares a variable @formatedDate to share the formatted string.
    2. It uses the CONVERT function with style 101 to format the date in the desired format(MM/DD/YYYY).
    3. It returns the formatted date string.

2. ReverseDate:

This
Function converts a date integer(YYYYMMDD format) into a “YYYYMMDD” string.

- Parameter:
    - @date:The date integer in YYYYMMDD format.
- Logic:

1. It declares a variable @reversedDate to store the formatted string.
2. It uses the CONVERT function with style 112 to convert the integer to a string in YYYYMMDD format.