Suppose you have a database named **dbERP** with the following two tables: **tblOrders** and **tblProducts**

Table 1: tblProducts

- intProductId (Identity Column, integer): Primary key for each product
- strProductName (nvarchar): Name of the product
- numUnitPrice (decimal): Unit price of the product
- numStock (decimal): Current stock quantity of the product

intProductId	strProductName	numUnitPrice	numStock
1	Widget A	50	300
2	Gadget B	75	150
3	Tool C	100	200

Table 2: tblOrders

- intOrderId (Identity Column, integer): Primary key for each order
- intProductId (integer, Foreign Key): References tblProducts.intProductId
- strCustomerName (nvarchar): Name of the customer who placed the order
- numQuantity (decimal): Quantity ordered
- dtOrderDate (datetime): Date of the order

intOrderId	intProductId	strCustomerName	numQuantity	dtOrderDate
1	1	John Doe	20	2024-01-15 14:00:00
2	2	Jane Smith	10	2024-02-10 10:30:00
3	1	Sam Wilson	15	2024-03-05 09:15:00

Task Requirements

API 01: Create a new order for an existing product. For example, place an order for "Tool C" by customer "Alex Brown" with a quantity of 25. Before creating the order, check if sufficient stock is available. If not, return a message indicating insufficient stock. After the order is created, deduct the ordered quantity from tblProducts.numStock.

API 02: Update an order's quantity. For example, update order ID 2 to increase the quantity to 15. Ensure that this change doesn't exceed the available stock. If there's enough stock, adjust the quantity in tblOrders and update the tblProducts.numStock accordingly.

API 03: Delete an order by intOrderId. Once an order is deleted, add the quantity back to tblProducts.numStock.

API 04: Retrieve all orders with the product details. This endpoint should return each order along with its strProductName and numUnitPrice.

API 05: Get a summary of total quantity ordered and total revenue for each product. The response should include:

- Product name
- Total quantity ordered
- Total revenue (calculated as numQuantity * numUnitPrice)

API 06: Retrieve all products with a stock quantity below a specified threshold (e.g., less than 100) and display their names, unit prices, and stock quantities.

API 07: Get the top 3 customers by total quantity ordered across all products.

API 08: Find the products that have not been ordered at all. This requires identifying records in tblProducts with no corresponding entries in tblOrders.

API 09: Implement a transactional operation for bulk order creation. Accept a list of orders, validate each one (for stock availability), and insert them as a single transaction. If any order fails (e.g., due to insufficient stock), roll back the entire operation.